

AD-A121 986

RIME: THE RECOVERABLE ITEM MANAGEMENT EVALUATOR VOLUME
II SECTION II STOC. (U) DECISION SYSTEMS DAYTON OH
W 5 DENNY MAY 80 TR-80-02-B F33600-78-C-0524

1/1

UNCLASSIFIED

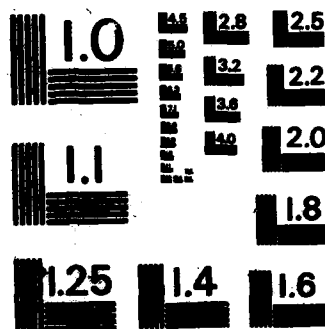
F/G 9/2

NL

END

FILMS

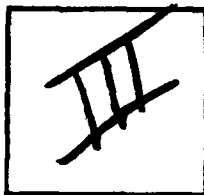
111



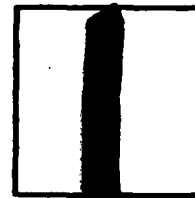
PHOTOGRAPH THIS SHEET

AD A121986

DTIC ACCESSION NUMBER



LEVEL



INVENTORY

RIME: Vol. II, Section II

DOCUMENT IDENTIFICATION

Ref. No. TR-80-02-B

Contract F33600-78-C-0524

May 80

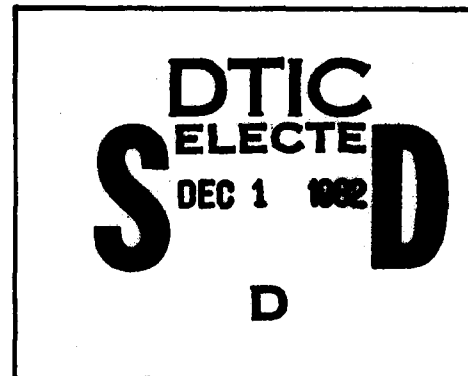
DISTRIBUTION STATEMENT A

Approved for public release
Distribution Unlimited

DISTRIBUTION STATEMENT

| | |
|--------------------|----------------------|
| ACCESSION FOR | |
| NTIS | GRA&I |
| DTIC | TAB |
| UNANNOUNCED | |
| JUSTIFICATION | |
| | |
| | |
| BY | |
| DISTRIBUTION / | |
| AVAILABILITY CODES | |
| DIST | AVAIL AND/OR SPECIAL |
| H | |

DISTRIBUTION STAMP



DATE ACCESSIONED



DATE RECEIVED IN DTIC

PHOTOGRAPH THIS SHEET AND RETURN TO DTIC-DDA-2

RIME:

The Recoverable Item Management Evaluator:

Volume II, Section II

Stock Level Computation Programs

RIME:

The Recoverable Item Management Evaluator:

Volume II, Section II

Stock Level Computation Programs

by

W. Steven Demmy

May 1980

TR-80-02-B

**DECISION SYSTEMS
2125 Crystal Marie Drive
Dayton, Ohio 45431**

(513) 426-8515

DISTRIBUTION STATEMENT A
Approved for public release
Distribution Unlimited

82 11 30 026

| REPORT DOCUMENTATION PAGE | | READ INSTRUCTIONS BEFORE COMPLETING FORM |
|---|-----------------------|---|
| 1. REPORT NUMBER | 2. GOVT ACCESSION NO. | 3. RECIPIENT'S CATALOG NUMBER |
| 4. TITLE (and Subtitle) RIME: The Recoverable Item Management Evaluator: Volume II: Program Listings and Narratives | | 5. TYPE OF REPORT & PERIOD COVERED INTERIM August 79-Jun. 80 |
| 7. AUTHOR(s) W. Steven Denny | | 6. PERFORMING ORG. REPORT NUMBER TR-80-02-B |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS Decision Systems 3575 Charlene Drive Dayton, Ohio 45432 | | 8. CONTRACT OR GRANT NUMBER(s) C-0524 F33600-78-B-0214 |
| 11. CONTROLLING OFFICE NAME AND ADDRESS 2750th ABW/PMA BLDG 1, Area C Wright-Patterson AFB, Ohio | | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 23041A5 |
| 14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) | | 12. REPORT DATE May 1980 |
| | | 13. NUMBER OF PAGES 298 |
| | | 15. SECURITY CLASS. (of this report) Unclassified |
| | | 16a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A |
| 16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited. | | |
| 17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) A | | |
| 18. SUPPLEMENTARY NOTES | | |
| 19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Recoverable item, multi-echelon, inventory/repair, simulation, METRIC, MOD-METRIC, AFLC | | |
| 20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This report describes the Recoverable Item Management Evaluator (RIME), a FORTRAN simulation model for evaluating the relative cost-effectiveness of analytic optimization procedures proposed for use in Air Force Logistics Command recoverable item management systems. Major features of the model include (a) the use of actual Air Force demand histories to drive the model demand processes, (b) modeling of current Air Force statistical estimation procedures, and (c) modeling of the dynamic interactions among initial provisioning, replenishment and distribution policies. Volume II documents the programs for RIME. | | |

FORM 1473

SECTION OF 1 NOV 68 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Section II

Levels Computation System Programs

List of Programs

Program

DMSGNI

EVALUATE

GETBSO

GETDAT

INTPRO

LEVLP1

LEVLP1

METINP

ONEIND

OVHSTL

SAVDAT

SPNDMS

TWOIND/INPUT

TWOIND/OUTPUT

Subroutine: DMSGN1

Function:

This routine generates input data sets for the Levels Computation System.

Description:

This routine reads D041 historical data for LRU/SRU data sets, and produces as output appropriate data cards required to drive MOD-METRIC computation routines. For a detailed discussion of this program, see the Levels Computation System Chapter in Volume I.

DMSGN1


```

05 C
06 C***** I.D
07 C
08 C READ SIMULATION PARAMETERS
09 C
10 WRITE(6,17) " NFORP, NLENGP,INOTR,NDNIS"
11 17 FORMAT(IV)
12 READ(5,13) NFORP, NLENGP,INOTR,NDNIS
13 WRITE(6,128) NFORP, NLENGP,INOTR,NDNIS
14 23 FORMAT(I9,4I7)
15 C
16 ITOT=NDNIS+INOTR
17 IF(ITOT.LE.16) GO TO 19
18 PRINT,"NDNIS+INOTR CANNOT EXCEED 16. STOP RUN"
19 STOP
20 C
21 C***** I.C
22 C READ WRITE FLAG CARD.
23 C
24 19 CONTINUE
25 WRITE(6,17) "WRITE FLAGS "
26 READ(5,13) IWT
27 WRITE(6,3) IWT
28 3 FORMAT(/"WRITE FLAGS=","5(9I2,4X))
29 C
30 INBUG=INT(1)
31 C
32 C***** I.D
33 C READ TITLE AND COMMENT CARD.
34 C
35 C
36 WRITE(6,13) "IDENT COMMENT"
37 ENCODE(TITLE,37) INHSH,INOHAS,ICOST,KHSH,KHOBAS,KCOST,
38 INHSH,BOHINI,KHINSK,BOHINI
39 37 FORMAT( 3I2," ",3I2," ",I2,I5.5," ")
40 8 I2,4I5(3)
41 WRITE(6,13) IDENT,TITLE
42 WRITE(1,69) IDENT,TITLE
43 WRITE(8,64) IDENT,TITLE
44 63 FORMAT("629,1X,I3,1X,"INITIAL PROV"," ",A40)
45 64 FORMAT("629,1X,I3,1X,"REPLN"," ",A40)
46 C
47 C***** I.E
48 C READ EVALUATOR PARAMETER DATA.
49 C
50 WRITE(6,17) "NORP, KBO, CFAC9"
51 READ(5,19) NORP,KBO,CFAC9
52 WRITE(6,13) NORP,KBO,CFAC9
53 WRITE(8,88) NORP,KBO,CFAC9
54 WRITE(1,88) NORP,KBO,CFAC9
55 68 FORMAT("969,1X,2I2,I5.2)
56 C

```

2 01 10-20-79 10.356 38

```

57 C
58 C*****I.F
59 C      READ OUTPUT CONTROL DATA
60 C
61 WRITE(6,13) NOUT,IPENT,IPNCH,IBSO
62 READ(5,13) NOUT,IPENT,IPNCH,IBSO
63 WRITE(6,13) NOUT,IPENT,IPNCH,IBSO
64 WRITE(8,13) NOUT,IPENT,IPNCH,IBSO
65 WRITE(1,13) NOUT,IPENT,IPNCH,IBSO
66 83 FORMAT('000,1X,127')
67 C
68 C*****I.G
69 C      READ OPTIMIZATION PARAMETERS.
70 C
71 WRITE(6,17) NBS,BETA,BSTART,BSTOP,CFAC,PSINC
72 READ(5,17) NBS,BETA,BSTART,BSTOP,CFAC,PSINC
73 WRITE(6,17) NBS,BETA,BSTART,BSTOP,CFAC,PSINC
74 WRITE(8,103) NBS,BETA,BSTART,BSTOP,CFAC,PSINC,KHNSK,BONNS,KNSAS
75 WRITE(1,103) NBS,BETA,BSTART,BSTOP,CFAC,PSINC,KHNSK,BONNS,KNSAS
76 103 FORMAT('000,1X,1315F6.2,161F6.4,16')
77 C
78 C*****I.H
79 C      READ FLYING HOUR PROGRAM AND GET DATA.
80 C
81 CALL NBSHA
82 C
83 C*****II.A
84 C      COMPUTE TOTAL FLYING HOURS PER BASE
85 C
86 PNAKPF=0.
87 STPH=0.
88 IS=NDNIS+1
89 IS=NDNIS+8
90 IF (IS.GT.23) IS=23
91 DO 120 IQ=IS,IN
92   TPN=0.
93   DO 110 KQ1=NBSNS
94     TPN=TPN+DPH(IQ,K)
95   110 CONTINUE
96   STPH=STPH+TPN
97   IF (PNAKPF.LV.TPN) PNAKPF=TPN
98 120 CONTINUE
99   AVTPH=STPH/FLOAT(IN-IS+1)
00   PNAKPF=PNAKPF/AVTPH
01 C      CONVERT QTN DATA TO HOURS
02   AVNPHR=AVTPH/3.
03   WRITE(6,18) STPH=0,STPH, AVNPHR=0,AVNPHR,
04   " PNAKPF=0,PNAKPF
05 C
06 C*****II.B      INITIALIZED RANDOM NO. GENERATOR AND
07 C      RECORD SEQUENCE NO: (1288)
08 C

```

T 01 10-20-79 10.366 28

```

09      R=RANDU(-.1)
10      ISEQ=0
11      C
12      C
13      C
14      C***** I2.C
15      C      SKIP THE FIRST (NPERD-1) LRU GROUPS
16      C
17      CALL READP1
18      NSKIP=NPERD-1
19      IF(NSKIP.LE.0) GO TO 60
20      DO 50 N=1,NSKIP
21      NGROUP=NG
22      CALL READPL(IENDP)
23      IF(IENDP.EQ.1) GO TO 1200
24      50 CONTINUE
25      60 CONTINUE
26      C
27      C      BEGIN GROUP LOOP
28      C
29      DO 100 N=NPERD,NLGRP
30      NGROUP=NG
31      WRITE(6,12)"BEGIN LRU GROUP=",NG
32      13 FORMAT(V)
33      C
34      C
35      C***** I2.A
36      C      SET LRU AND SRU HEADEN DATA
37      C
38      CALL READPL(IENDP)
39      IF(IENDP.EQ.1) GO TO 1200
40      C
41      C      START PROCESSING RECORDS.
42      C
43      C-----+----- FILE SECTION 2-----+-----
44      C
45      C      ZNOTE= THE NUMBER OF PERIODS OF DATA TO BE
46      C      COMPUTED.
47      C
48      C      NDNIS = INITIAL NUMBER OF PERIODS USED IN
49      C      ESTIMATE.
50      C
51      C-----+-----+-----+-----+-----+-----+-----+-----+-----+-----
52      C
53      C***** I2.B
54      C      COMPUTE OST BY BASE USING LRU OST TIME AS THE GRAND
55      C      AVERAGE.
56      C
57      OSTLRU=OST(1)
58      AVEOST=0.
59      WRITE(6,13)" OSTLRU=",OST(1)
60      DO 15 N=1,NBASES

```

T 01 10-20-79 10.366 JS

```

61      OST(K)=OST(K)+OSTDLT(K)
62      IF(OST(K).LT.5.)OST(K)=5.
63      AVOST=EVOST+OST(K)
64      150 CONTINUE
65      AVOST=EVOST/FLOAT(NBASES)
66      C
67      C
68      C***** III.C
69      C      BEGIN QUARTER LOOP
70      C
71      DO 500 IQTR=1,INQTR,2
72      C
73      IF(IMBUS.GE.1)WRITE(6,153)IQTR,NV
74      153 FORMAT(/,15('=-='),"START QTR=",I3,
75      " FOR GROUP=",I3,5('=-=')/)
76      C
77      C
78      C***** III.C,1
79      C      SET AVERAGE FUTURE FLYING HOURS PER QUARTER
80      C      USING AVERAGE OF THE NEXT 8-QUARTERS.
81      C
82      TOPH=0.
83      DO 177 N=1,NBASES
84      BPHR(K)=0.0
85      IS=NDNIS+QTR
86      IN=MIN(NDNIS+QTR+7,23)
87      DO 176 I=IS,IN
88      176 BPHR(K)=BPHR(K)+BPH(I,K)
89      BPHR(K)=BPHR(K)/FLOAT(IN-IS+1)
90      177 TOPH=TOPH+BPHR(K)
91      AVHACH=TOPH/J.
92      C
93      IF(IMBUS.GE.1)WRITE(6,173)TOPH,IS,IN,(BPHR(K),K=1,NBASES)
94      173 FORMAT(/,9 TOPH="F10.0" = TOTAL PROJECTED AIRCRAFT
95      " PROGRAM FOR QTR ",I2," THRU QTR ",I2/
96      " PROGRAM BY BASE = BPHR(K) ="F10.0)
97      C
98      C
99      C***** III.C,2
100     C      COMPUTE AVERAGE LRU INSTALLED PROGRAM
101     C      IN THE NEXT 8 QUARTERS.
102     C
103     IPHAKP=0
104     IPROG=0
105     IS=MIN(NDNIS+QTR,16)
106     IN=MIN(NDNIS+QTR+7,16)
107     DO 178 I=IS,IN
108     EX=IPPROG(I,4)
109     IF(IPHAKP.LT.EX)IPHAKP=EX
110     IPROG=IPROG+EX
111     178 CONTINUE
112     C

```

T 01 10-20-79 10.366 JS

```

113 C***** IIX.C.3
114 C      CONVERT PROGRAM FROM 1000S OF HOURS TO HOURS.
115 C      THEN ESTIMATE HOURS/ MONTH ON THE HOST AIRCRAFT
116 C
117      IPRG= 100 * IPRG
118      IPEAKP=100 * IPEAKP
119 C
120      AVQFHR=FLOAT(IPRG)/FLOAT(IE410+1)
121 C      ESTIMATE FLYING HOURS RELATED TO THIS PROGRAM
122 C
123      PEAKPF=FLOAT(IPEAKP)/AVQFHR
124      AVQFHR=AVQFHR/FLOAT(IOBA(1))
125      AVMFHR=AVQFHR/3.
126      IF (INBUS,GE,1)WRITE(6,8173) ISYIE,IPRG,AVQFHR,AVMFHR,PEAKPF
127 8173 FORMAT(1X, ' AVERAGE LRU PROGRAM CALCULATIONS' /
128 &      T2, "IS="I2, " IE="I2, " IPRG="I10,
129 &      " AVQFHR="F10.0, " AVMFHR="F10.0, " PEAKPF="F6.3)
130 C
131 C***** IIX.C.4
132 C      COMPUTE FUTURE MONTHLY INSTALLED PROGRAM
133 C      BY BASE PROPORTIONAL TO AIRCRAFT FLYING PROGRAMS.
134 C      IF INSTALLED BASE PROGRAM IS LESS THAN 1.0,
135 C      RESET IT TO 1.0.
136 C      THIS IS TO PREVENT DIVIDE AND LOOPING ERRORS
137 C      IN THE MOD-METRIC ROUTINES.
138 C
139 C
140      AVBFHR=0.
141      DO 179 R=1,NBASES
142      FH(K)=AVBFHR+(BFHR(K)/TOPH)
143      IF(FH(R).LT.1.)FH(K)=1.0
144      AVBFHR=AVBFHR+FH(K)
145 179 CONTINUE
146      AVBFHR=AVBFHR/FLOAT(NBASES)
147 C
148      WRITE(6,183)IOTR,AVBFHR,(FH(K)K=1,NBASES)
149 183 FORMAT(//9 '*****UNEGN1=-IOTR*E12, ' AVE BASE FLYING',
150 &      " PROGRAM FOR AIRCRAFT OVER NEXT 6-QTRS =",
151 &      F10.2//T20,PROGRAM BY BASE * FH(K)="BFH.2)
152 C***** IIX.C.5
153 C
154      IF(IOTR.GT.1) GO TO 180
155 C
156 C      SET PARAMETERS FOR INITIAL PROVISIONING CALCULATIONS
157 C
158      IFILE=1
159      JEQBAS=IEQBAS
160      JMETH=IMETH
161      JCOST=ICOST
162      GO TO 184
163 C
164 C      SET PARAMETERS FOR REPLENISHMENT CALCULATIONS

```

2 01 10-20-79 10.366 25

```

165 C
166 180 CONTINUE
167 IFILE=8
168 JEQDAS=NEQDAS
169 JHETH=KHETH
170 JCOST=KCOST
171 C
172 C*****IX.C.6
173 C CONSTRUCT PROGRAM PLANNING DATA
174 C
175 184 CONTINUE
176 C
177 C IF JEQDAS=1 THEN OUTPUT A "97" RECORD WITH
178 C FH(K),K=1,...,NBASES SET EQUAL TO AVEFNR, AND
179 C OST(K),K=1,...,NBASES SET EQUAL TO AVEOST.
180 C OTHERWISE, USE ORIGINAL FH AND OST ARRAYS
181 C
182 C
183 C*****IX.C.6.1
184 C WRITE OUT FLYING HOUR PROGRAM AND OST DATA
185 C
186 C
187 IF(NBASES.LE.8) GO TO 79
188 73 FORMAT("979,1X,I3,6(F6.0,F3.0)/2"97",T7.
189 & 6(F6.0,F3.0))
190 C
191 WRITE(IVIDE,73)
192 & BASES,((FH(K),OST(K)),K=1,NBASES)
193 GO TO 81
194 C
195 74 FORMAT("979,1X,I3,6(F6.0,F3.0))
196 C
197 79 CONTINUE
198 WRITE(IVIDE,74)
199 & BASES,((FH(K),OST(K)),K=1,NBASES)
200 C
201 C*****IX.C.6.2
202 C COMPUTE QUANTERS TO BE USED FOR PLANNING FACTOR CALCULATIONS
203 C ALWAYS USE AT LEAST 4 QUANTERS OF DATA IN
204 C COMPUTING PLANNING FACTORS
205 C
206 C
207 81 CONTINUE
208 IS=MAX(4,NDNIS+IQTR-8)
209 IS=NDNIS+IQTR-1
210 IS=MIN(88,IS)
211 IS=MAX(8,IS)
212 C LOOP THRU N NUMBER OF ITEMS:
213 75 DO 100 J=1,NITEMS
214 C*****IX.C.6.3.A
215 C WRITE OUT TO FILE#7 THE LRU/SRU PAIRLT RECORDS.
216 C IDENTIFY LRUs AND SRUs FOR IC1

```


ST 01 10-20-79 10.366 JS

```

117      IC2=1
118      IC1=3
119      IF(J.EQ.1) IC1=2
120      IF(JMETH.EQ.3) IC1=1
121      IF(JMETH.EQ.4) IC1=1
122      C
123      C      JMETH=1 INDICATES OWNED RUN.
124      C
125      IF(JMETH.EQ.1) IC1=1
126      IF(JMETH.EQ.1) IC2=1
127      C
128      C***** III.C.6.3.B
129      C      COMPUTE PLANNING FACTORS--XMTBD,XMRTS,CONS
130      C
131      IPT=J
132      CALL METEND(XS,IE,IPT)
133      C
134      C***** III.C.6.3.C
135      C      COMPUTE PROCUREMENT LEADTIME IN MONTHS
136      IFLT(J)=(LTADN(J)+LTPROD(J))/30
137      C
138      C
139      C      SCALE PARAMETERS FOR OUTPUT IN MODMETRIC FORMAT
140      C      ROUNDING ALL CALCULATIONS.
141      C***** III.C.6.3.D      SCALE PARAMETERS FOR OUTPUT IN
142      C      MODMETRIC FORMAT
143      C
144      ICON1=IFIX(100.0*CONS(J)+0.5)
145      XMRTS1=IFIX(100.0*XMRTS(J)+0.5)
146      XMTBD1=IFIX(XMTBD(J)+0.5)
147      ICOST=IFIX(UCOST(J)+0.5)
148      IDRCD=IDDT(J)+IDRT(J)
149      C
150      C***** III.C.6.3.E
151      C      COMPUTE OVERHAUL STOCK LEVEL
152      C
153      IPOTR=IOTR
154      JJ=J
155      CALL OVHSTA(IPOTR,JJ,IYSTL)
156      C
157      C      IF JMETH IS NOT EQUAL TO 4.
158      C      SKIP THE XPMCR $7-27 CALCULATION.
159      IF(JMETH.EQ.4) GO TO 300
160      C
161      C***** III.C.6.3.F
162      C      SET PARAMETERS FOR INITIAL PROVISIONING CALCULATIONS
163      C
164      IF(IOTR.EQ.1) GO TO 300
165      DRCD=IDRT(J)
166      DRCD=IDRCD
167      FLT=IFLT(J)
168      OPA=IOPE(J)

```

2 01 10-20-79 13.366 JS

```

169 C
170 C      STATE INITIAL PROV. PROGRAM DATA IN 100'S OF HOURS
171 C
172 C      AVHPPG=SYNTH/100.
173 C      SHAKPG=SHAK/100*AVHPPG
174 C
175 C      CALL INTPRO(PEN(J),AVHPPG,UCOST(J),XHTD(J),XHTS(J),
176 C      CONS(J),DECD,DECDDELTA,QDA,PEN(0),AVHPPG,
177 C      SHAKPG,ITTR)
178 C
179 C      WRITE ITTR DATA
180 C
181 C      300 CONTINUE
182 C
183 C*****
184 C      IF JCOST.EQ. 1. APPLY UNIT COST DISCOUNT
185 C
186 C      IF(JCOST.EQ.1) GO TO 310
187 C      DISCNT=1-XHTS(J)
188 C      IF(DISCNT.LT.0.10)DISCNT=0.10
189 C      UCOST=UCOST*(1-DISCNT)
190 C      310 CONTINUE
191 C*****
192 C      III,C,6,3,N OUTPUT HOENETRIC ITTR DATA RECORD
193 C
194 C      LAMBDX=0
195 C      JOBID=100000000*IDENT+1000000*HAMBDA+10000*NG+100*ITTR+J
196 C      WRITE(1FELB,43) IC1,IC2,JOBID,ISVOTL,UCOST,
197 C      INHDD1,INHDS1,IC0N1,IOQA(J)
198 C      INHDD1,INHDS1,INHDS1,INHDS1
199 C      43 FORMAT(2I1,1X,2I10,1X,2I3,3X,2I5,2I6,1X,2I3,2I3)
200 C
201 C*****
202 C      WRITE DELIVERY SCHEDULE CARD
203 C
204 C      IC2=5
205 C      IF((JNTH.NE.0).AND.(ITTR.NE.1))
206 C      WRITE(1FELB,109)IC1,IC2,JOBID,
207 C      ISVOTL,ITTR
208 C      109 FORMAT(2I1,1X,2I10,1X,2I3,1X,2I5,2I6,1X,2I3,2I3)
209 C
210 C      END OF ITTR LOOP
211 C
212 C      100 CONTINUE
213 C
214 C      END OF ONE LOOP
215 C
216 C      500 CONTINUE
217 C
218 C      END OF GROUP LOOP
219 C      READ NEXT LNU/GRU FAMILY.
220 C
221 C      1000 CONTINUE
222 C      WRITE(6,13)"DNSSN2 SUCCESSFUL COMPLETION"

```

ST 01 10-20-79 10.366 28

521 STOP

522 1200 CONTINUE

523 WRITE(6,13)='END OF FILE 07 REACHED, STOP RUN'

524 WRITE(6,13)='LAST LRU READ WAS RUN='.PSN(17

525 STOP

526 END

*W 7 MEMORY EXPANDED. USE SLIMITS OR CONN= OPTION FOR NEXT RUN

Subroutine: EVALUATE**Function:**

This program determines an optimal distribution of a given number of LRU/SRU assets.

Description:

The program EVALUATE determines an optimal distribution of stock among the depot and using bases. A discussion of the functions of this routine and input requirements is presented in AFLCR 57-13, and will not be discussed further here. The routine utilized by RIME is identical to the routine discussed in AFLCR 57-13 with one exception. Provisions have been added to the RIME version to permit output of computed levels to the file IBSO using the same record formats as output by the program GETBSO. Specific program lines that were changed for the RIME version are 421-425; 1295-1315; and 3581-3675. The portions of changed code are marked by comment statements in the text of the program.

T 02 10-20-79 10.541 EVALUATE

```

53      ZSW1=0
54      ZSWLNU=0
55      ZSWSEU=0
56      MAXSEU=49
57      ZED=30
58      ZETA43=0
59      ZYBHX=0
60      ZYCOST=0
61      C
62      6  READ(577,ZYD=9907IC1,ZC2,CARD
63      7  FORMAT(2E1,1E,AGE)
64      8  IF(INOZ,ZZ,0)GO TO 2100
65      GO TO 110,20,30,30,6,6,6,6,907,IC1
66      C
67      90 IF(ZSWSEU,ZZ,0)GO TO 1000
68      IF(ZSW1,ZZ,0)GO TO 1000
69      GO TO 10,92,6,6,6,96,97,98,612IC2
70      C
71      92 DECODE(CARD,192)NDC
72      192 FORMAT(18A4)
73      ZND=0
74      GO TO 6
75      C
76      96 Z96=1
77      ZND=0
78      DECODE(CARD,196)NOTR,Z97,CPC
79      196 FORMAT(2E2,7E,2)
80      IF(NOTR,ZZ,1)NOTR=0
81      IF(CPC,ZZ,1)CPC=0
82      GO TO 6
83      C
84      97 Z97=1
85      ZND=0
86      DECODE(CARD,197)NBASIS
87      197 FORMAT(1E)
88      ZB=1
89      ZB=ZB*(6,NBASIS)
90      297 DECODE(CARD,297)(PH(I),OST(ZYET=ZB,18)
91      397 FORMAT(3E,6(96,0,73,0))
92      IF ( LB,ZZ,NBASIS ) GO TO 567
93      ZB=ZB+6
94      ZB=ZB*(6,NBASIS)
95      READ(577,IC1,IC2,CARD
96      IF(IC1,ZZ,9,AND,IC2,ZZ,7)GO TO 297
97      496 WRITE(6,997)
98      497 FORMAT(13HFLYING HOUR CARD MISSING/INCOMPLETE)
99      STOP
100     597 ADDRE = 0.0
101     PLOTOT = 0.0
102     ONIN = 1E35
103     ONAX = 1E-35
104     DO 697 I=1,NBASIS

```


2 02 10-20-79 10.341 EVALUATE

```

55      DO 315 K=1,NB
56      315 ONLRO=ONLRO+OBL(1,K)
57      IF(K=NB,0)GO TO 6
58      DO 415 K=1,NB
59      415 ONLNU=ONLNU+LPP(1,K)
60      GO TO 6
61      C
62      16 DECODE(CARD,116)NSH
63      116 FORMAT(25X,6A4)
64      GO TO 6
65      C
66      C
67      20 GO TO 121,6,6,6,25,26,6,6,6,6,2
68      C
69      21 IF(ISHW,0)GO TO 1000
70      IF(ISH(1,SH,0)0)GO TO 1000
71      ISWLU=1
72      DECODE(CARD,111)IDLU,C,INTEDTYN25,CQULRU,NAP2.
73      ONLNU=DATE,PLZL
74      INRC = INRC + 1
75      IF(INRC,0)NAP2=1.
76      DO 221 K=1,NBASS
77      221 IF(K=NB(K)*NAP2/(INTED=30.Y
78      J=0
79      NSRU=0
80      PLZ=PLZL
81      ONLNU = DATE + KDDNT
82      CONLCARD
83      GO TO 111
84      C
85      25 DECODE(CARD,125)XD
86      125 FORMAT(6A4)
87      DO 1125 I = 1,6
88      1125 IF(ID(I).NE. IDRU(I))GO TO 6
89      DECODE(CARD,125)OBL,EP
90      225 FORMAT(25X,(4X2,1X,4X2),1X,6X)
91      DO 325 K=1,NB
92      325 ONLRO=ONLRO+OBL(1,K)
93      IF(K=NB,0)GO TO 6
94      DO 425 K=1,NB
95      425 ONLNU=ONLNU+LPP(1,K)
96      GO TO 6
97      C
98      26 DECODE(CARD,126)NSH
99      126 FORMAT(25X,6A4)
100     GO TO 6
101     C
102     30 GO TO 131,6,6,6,35,36,6,6,6,6,2
103     C
104     31 IF(ISHW,0)GO TO 100
105     ISJ=1
106     NSRU=NSRU+1

```


: 02 10-20-79 10.841 EVALUATE

```

07      ZSWSHU=1
08      IF (NSTR,GT,NAXSHU) GO TO 2200
09      DECODE(CARD,111)(IDSHU(J,X)X1,0),CASH(J),SHRSH(J),SHRSH(0)
10      * CONSHU(J),SAPSH(J),
11      * SHRSH(J),DRSH(J),SPLTS(J)
12      C
13      IF (SAPSH(J),LT,1)YSHAPSH(J)=1.
14      IF (SHRSH(J),EQ,0)SHRSH(J)=000000
15      IF (SHRSH(J),EQ,0)SHRSH(J)=250
16      IF (SHRSH(J),EQ,0)SHRSH(J)=650
17      SHRSHU(J) = SHRSH(J) + ADHSH
18      CON(J)=CONSHU(J)
19      SPLT(J)=SPLTS(J)
20      IF (SHRSH(J),GT,0) GO TO 231
21      CON(J)=CON(J)+SHRSH(J)
22      SPLT(J)=CON(J)+SPLT(J)+301+0PAC/SHRSH(J)
23      SHRSHU(J)=SPLT(J)+(SHRSH(J)+CON(J))+SHRSHU(J)/SHRSH(J)
24      GO TO 6
25      231 SHRSHU(J)=CON(J)
26      SPLT(J)=60.
27      SHRSHU(J)=60.
28      GO TO 6
29      C
30      35 DECODE(CARD,135)XD
31      135 FORMAT(6A4)
32      DO 135 K=1,6
33      1135 IF (XD(K),.F)ZSHU(J+K)YUO TO 6
34      DECODE(CARD,135)(OSH(J,X)X10100)1(S50(J,X),X=1.07
35      235 FORMAT(25X,14I2,1X,4I2),1X,0XV)
36      DO 235 K=1,NSTR
37      335 OSHU(J)+OSHSHU(J7+OSH(J,X))
38      IF (K=7,NO,0)GO TO 6
39      DO 435 K=1,KSP
40      435 OSHSHU(J)+OSHSHU(J7+SHRSH(J,X))
41      GO TO 6
42      C
43      36 DECODE(CARD,136)(SHSH(J,X),X=9.07
44      136 FORMAT(25X,4A4)
45      GO TO 6
46      C
47      990 ZSHU=1
48      IF (ZSWSHU,NH,0)GO TO 1000
49      IF (ZSWSHU,NH,0)GO TO 1000
50      GO TO 8100
51      C
52      1000 IF (ZSHU,NH,1)AND ZSHU,NH,1 (AND, ZSHU,NH,1)GO TO 1020
53      WRITE(6,1010)ZSHU,NH,1
54      1010 FORMAT(11I20M,12.5X,4I20V=1X,5X,4I20V=1X,10X,5I20V=1X)
55      STOP
56      C
57      1020 ZSWSHU=0
58      ZSWLNU=0

```

[illegible]

T 02 10-20-79 10.561 EVALUATE

```

11 1055 IF (NEND.00.0) GO TO 2060
12 C
13 WRITE (NOUT,1057)
14 1057 FORMAT(4H SRU)
15 DO 1060 N=1,NBSRU
16 C-----U-----WRITE SRU ITEM DATE
17 WRITE (NOUT,1055)K,(IDSRU(K),STRT(K),CONRU(K),ONSRU(K),SHRSD(K),
18 *SNTS(K),CONSRU(K),DTSRU(K),SNTS(K),SKPR(K),SPITS(K)
19 * (SNTSD(K),ST.0)
20 IF (ONSRU(K),ST.0) WRITE (NOUT,3007)
21 3007 FORMAT(12, "***** WARNING - SRU STOCK LESS THAN ZERO")
22 IF (DTSRU(K).LE.0) WRITE (NOUT,3002)
23 3002 FORMAT(12, "***** WARNING - SRU LESS THAN OR EQUAL TO ZERO")
24 IF (DTS(K).LE.0) WRITE (NOUT,3004)
25 3004 FORMAT(12, "***** WARNING - SRU LESS THAN OR EQUAL TO ZERO")
26 IF (SNTSD(K).GE.1760 TO 3160
27 WRITE (NOUT,3006)
28 3006 FORMAT(12, "***** FATAL ERROR ***** - NTRD LESS THAN ONE")
29 STOP
30 3100 IF (SHRSD(K).GE.0 .AND. SNTS(K).LE.1) GO TO 3110
31 WRITE (NOUT,3008)
32 3008 FORMAT(12, "***** FATAL ERROR ***** - NTRD IS INCORRECT")
33 STOP
34 3110 IF (CONSRU(K).GE.0 .AND. CONRU(K).LE.1) GO TO 3120
35 WRITE (NOUT,3010)
36 3010 FORMAT(12, "***** FATAL ERROR ***** - CONDEMNATION IS INCORRECT")
37 STOP
38 3120 IF (CONRU(K).GT.0 .AND. SNTS(K).ST.1760 TO 3160
39 GO TO 1060
40 3140 WRITE (NOUT,3150)
41 3150 FORMAT(12, "***** FATAL ERROR ***** -
42 * 1X, "CONDEMNATION > 0 AND FLT < 1")
43 STOP
44 1060 SNTSD(K)=SNTSD(K)/SNTS(K)
45 1065 FORMAT(12, "E3.2X,6A4,F9.0,X5F7.0,F9.1,5X,F4.2,2X,F6.2,1X,F2.0,
46 * 1X,E3.0,2X,F2.0,2X,F6.0,2X,6A4)
47 C
48 WRITE (NOUT,2000) (K,K=1,NBSRU)
49 2000 FORMAT(20H0 SRU FILL RATS BY NBSRU/7X,2360)
50 DO 2030 J=1,NBSRU
51 SDDSD(J)=
52 DO 2010 N=1,NBSRU
53 2010 SDDSD(J)=SNTS(J)*PH(K)*NBSRU/(SNTSD(J)*30.)
54 DO 2020 N=1,NBSRU
55 2020 DTRT(K)=DTSRU(J)
56 C
57 2030 CALL NEND(SDDSD,DTSRU(J),ONSRU(J),OSDSRU(J),
58 * SNTSD(J),SNTS(J),SKPR ,STOCK,J+1)
59 DO 2050 N=1,NBSRU
60 ADLST(N)=0
61 DO 2040 J=1,NBSRU
62 2040 ADLST(N)=ADLST(K)+BACK(K,J)

```

T 02 10-20-79 10.561 EVALUATE

```

63 2050 ADULT(N)=ADULT(K7/(TD(K)*41.-RNRST))
64 DO 2056
65 2050 DO 2055 J=JONES+1, NRASNS
66 2055 ADULT(N2055)=0.
67 2055 CONTINUE
68 DDDEL=0.
69 DO 2070 N=1, NRASNS
70 DDDEL=DDDEL+ENRST*TD(K)*PRD
71 2070 ENRST(K)=ENRSTLU+ADULT(K)
72 C
73 CALL NHRG(DDDEL, DDTLU, ONHRG, OPDLRU,
74 * YRTRD, YRSTS, BAK , STOCKS 1 )
75 BOM=0.
76 DO 2086 N=1, NRASNS
77 2080 BOM=0+BAK(K,1)
78 KCONV=175042(OPDLRU)+.997*52TDDTLU
79 KLSNS=KCONV-LCONV
80 ENRST(ENRST,2500)(K,K=1, NRASNS)
81 2090 FORMATT/7/6X,37NOPTIMAL DETERMINATION OF ON-HAND STOCK/1
82 * 5X,4NWORK,23X,5NTOAL,17X,5NRASNS/
83 * 5X,4NENRST,23X,2ONON-HAND ENRST CON,36X3)
84 C-----
85 ENRST(ENRST,2500)ENRSTLU,ONHRG,KSDRST,LCONV,LSSTOCK(K),K=1, NRASNS)
86 2100 FORMATT1600 LRU ,6A4,15,8X712/5X,23,2313)
87 C-----
88 IF(ISSO,0)WRITE(ISSO,2103)LINE,15LRU7CSTLRU,
89 * ONHRG,KSDRST,LCONV,LSSTOCK(K),K=1, NRASNS)
90 LNR=ISSO+1
91 2103 FORMATT(1X,15,"11",1X,6A4,15,8X712/5X,23,2313)
92 TCONV=0. ONHRG
93 C-----
94 IF(NRST,0)01000 2140
95 WRITE(ENRST,2110)
96 2110 FORMATT(5X,5X)
97 DO 2126 J=1, NSRU
98 CSTSRU=CSTSRU(J)*ENRST(J)
99 TCONV=TCONV+CSTSRU(J)*ONHRST/J
100 KCONV=175042(OPDSRU(J)+.997*52LX(J)/DHTSRU(J)
101 KSDRST=KCONV
102 2120 WRITE(ENRST,2130)J,(ISSRUS,2121=1.6),ONHRST(J,KSDRST,
103 * KCONV,LSSTOCK(K,J),K=1, NRASNS)
104 2130 FORMATT(2X,12,2X,6A4,15,8X712/5X,23,2313)
105 C-----
106 IF(ISSO,0)WRITE(ISSO,2103)LINE,(SDSRU(J,X),15,1.67,
107 * CSTSRU,ONHRST(J,KSDRST,LCONV,LSSTOCK(K,J),K=1, NRASNS)
108 LNR=ISSO+1
109 C-----
110 2140 WRITE(ENRST,2150)BOM,TCONV
111 2150 FORMATT110,25X,21NEXPERTEB NAKORDERS 0,13,0/
112 * 26X,12NTOAL COST 0,9X71250)
113 C
114 SYSSKK=STSRAN+BOM

```


Subroutine: GETBSO

Function:

This program records up to 20 sets of LRU/SRU stock levels, with each set of levels corresponding to a different Buy Support Objective (BSO). It then scans this set of points to identify the particular set closest to a given BSO.

Description:

This program inputs up to 20 sets of LRU/SRU stock levels, with each set corresponding to a different Buy Support Objective (BSO). It then scans these points, and determines the particular point closest to a given desired Buy Support Objective, and outputs the data set of levels associated with that point. A discussion of the computations performed by this routine and input and output records is given in AFLC 57-13, and will not be discussed further here.

The RIME version of GETBSO is identical to the AFLCR 57-13 routine with one exception. In program lines 2120-2200 logic has been inserted which replaces all blanks in the part number identification field with zeros. This is to facilitate sorting steps which are performed at a later stage of Levels Computation System job stream.

T 01 10-20-79 13.997 JS

```

1  *RUN=IRINE/ONE/GETBSO.O(BCD.WO60Y
2  *GETBSO.S
3  C      GETBSO
4  C      COMPILED FILE IS MODNETRIC/ONESS
5  C
6  C
7  C      READ AND SCORE(TEMPORARY WORK FILE) ALL DATA
8  C      EQUAL TO THE SAME WORK UNIT CODE(WUC).
9  C      THERE MAY BE UP TO 20 SETS PER WUC. EACH SET
10 C      IS IDENTIFIED BY AN "SS" CARD. THE DATA CARDS
11 C      FOLLOWING EACH "SS" CARD BELONG TO THAT SET.
12 C      EACH SET HAS A UNIQUE BACKORDER,CONT & BSO.
13 C      EACH SET EQUATES TO A POINT ON THE GRAPH.
14 C
15 C
16 C
17 C      I
18 C      I
19 C      I
20 C      I
21 C      IX (A)
22 C      I
23 C      I      X (B)
24 C      I
25 C      I      X (C)
26 C      I
27 C      I      X (D)
28 C      B I
29 C      I      X (E)
30 C      I
31 C      I      X (F)
32 C      I
33 C      I      X (G)
34 C      I
35 C      I      X (H)
36 C      I      X (I)
37 C      I      X (J)
38 C      I I I I I I I I I I I I I I I I I I I I I I
39 C      C
40 C      B = BACKORDERS      C = COST
41 C
42 C
43 C      ANALYZE ALL SETS(POINTS). DISCARD THE POINTS
44 C      THAT ARE NOT VALID
45 C      EX: POINTS "B", "D", & "G" ARE NOT ACCEPTABLE.
46 C      THEREFORE THEY ARE "NOSSON".
47 C
48 C      SEARCH ALL VALID POINTS FOR THE POINT CLOSEST TO
49 C      THE DESIRED BUY SUPPORT OBJECTIVE(BSBOY AND
50 C      OUTPUT THE DATA FOR THAT POINT
51 C
52 C

```

GETBSO


```

105 C
106 REWIND INH
107 C
108 C INCREMENT LAMBDA COUNTS
109 C
110 KNTB50=KNTB50+1
111 C
112 WRITE(ROUT,17)LINE,KNTB50,DESB
113 17 FORMAT(12,14,A150,"KNTB50 =",I3,"REMOVED",I3,
114 12,5)
115 LINE=LINE+1
116 WRITE(ROUT,18)LINE
117 18 FORMAT(12,14,A150,"37X,"Y D C BAKED")
118 LINE=LINE+1
119 IPT=0
120 C
121 C PROCESS FIRST "88" CARD AND RELATED DATA
122 C
123 20 READ(INH,25)IC,CARD
124 25 FORMAT(12,12,1X,A150)
125 IF(IC,88,88)GO TO 20
126 IPT=IPT+1
127 DECODE(CARD,87)DSO(IPT),WUC,WUNB,IPT,C(IPT)
128 C DSO - BUY SUPPORT OBJECTIVE
129 C WUC - WORK UNIT COST
130 C WUNB - NUMBER OF DATA LINES FOLLOWING 88 CARD
131 C B - BACKORDERS
132 C C - COST
133 27 FORMAT(12,5,A5,I3,F12,5,F12,5)
134 30 SWUC=WUC
135 WRITE(9)IC,CARD
136 C
137 DO 40 I=1,WUNB
138 READ(INH,25)IC,CARD
139 WRITE(9)IC,CARD
140 40 CONTINUE
141 IPT=IPT+1
142 C
143 C STORE ALL INPUT DATA PERTAINING TO THE SAME WUC
144 C WHEN NEW WUC IS FOUND PROCESS STORED DATA
145 C
146 200 READ(INH,25,WUNB=60)IC,CARD
147 IF(IC,88,88)GO TO 200
148 DECODE(CARD,87)DSO(IPT),WUC,WUNB,IPT,C(IPT)
149 IF(WUC,IN,SWUC)GO TO 90
150 WRITE(9)IC,CARD
151 C
152 205 DO 50 I=1,WUNB
153 READ(INH,25)IC,CARD
154 50 WRITE(9)IC,CARD
155 IPT=IPT+1
156 GO TO 200

```

```

57      C
58      60 INWD61
59      IPT=IPT-1
60      60 TO 100
61      C
62      C      HOLD NEW VUC DATA
63      C
64      90 HISTOZIT
65      IPT=IPT-1
66      NO=NO+1
67      SWUCWUC
68      CARDS=CARD
69      IC=IC+1
70      C
71      C      FIND THE GOOD SLOPES AND FLAG THE BAD POINTS
72      C
73      IF(IPT.EQ.1) GO TO 136
74      100 DO 135 I=3,IPT
75      NM=1
76      108 IF(I+NM.LE.3)GO TO 135
77      IF(FLAG(I+NM).NE.NOGOOD)GO TO 309
78      S1=(B(I+NM-1)-B(I+NM)) / (C(I+NM)-C(I+NM-1))
79      S2=(B(I+NM)-B(I)) / (C(I)-B(I+NM))
80      IF(S2.GT.S1)GO TO 134
81      FLAG(I+NM)=NOGOOD
82      109 NM=NM+1
83      60 TO 108
84      134 BSO(I)=S2
85      135 CONTINUE
86      C
87      136 CONTINUE
88      IF(IPT.LE.1) GO TO 306
89      C
90      C      SEARCH FOR THE BSO THAT IS DESIRED
91      C
92      DO 300 I=2,IPT
93      IF(FLAG(I).NE.NOGOOD)GO TO 309
94      IPT=I-1
95      IF(BSO(I).GT.CDSO)GO TO 303
96      300 CONTINUE
97      C
98      303 IF(FLAG(IPT).NE.NOGOOD)GO TO 304
99      IPT=IPT-1
100     GO TO 303
101     304 NEWIND J
102     IF(IPT.EQ.1)KST=1
103     DO 310 X=1,KST
104     305 READ(3)IC,CARD
105     IF(IC.NE.99)GO TO 303
106     310 CONTINUE
107     DECODE(CARD,27)E1,D2,NUMSTXNNW,CT
108     XBACK=XBACK+XNNW

```

DT 01 10-20-79 10.001 5

```

109      TCT=TCT+CT
110      C
111      DO 350 I=1,NUMB
112      READ(3)IN,CARD
113      LINE=LINE+1
114      DECODE(CARD,320)OUT1,FSW,OUT2
115      320 FORMAT(A20,A10,A00)
116      C-----END LINE MODIFICATIONS
117      DECODE(OUT1,323)XDRUM,LANDBASELNUV,XUTYNUV,NST
118      323 FORMAT(5X2,A00)
119      IVAL=100000000*XDRUM+1000000000*LANDBASELNUV+10000*XUTYNUV+100*XST+NUV
120      WRITE(OUT,326)XINH,IC,IVAL,NST
121      326 FORMAT(1X,I4,1X,I4,1X,I10,A4,A00)
122      IFLAG=1
123      IF(FLAG,NO:1) GO TO 350
124      C-----END LINE MODIFICATIONS
125      C
126      WRITE(OUT,325)LINE,IC,OUT1,OUT2
127      325 FORMAT(1X,I4,1X,I4,1X,A20,A00)
128      LINE=LINE+1
129      IC=IC+8
130      WRITE(OUT,330)LINE,IC,OUT1,FSW
131      330 FORMAT(1X,I4,1X,I4,1X,A20,4X,A10)
132      350 CONTINUE
133      C
134      IPT=1
135      REWIND 3
136      IF(INH,NO:1)GO TO 800
137      BSO(IPT)=BSO(NIPT)
138      S(IPT)=S(NIPT)
139      C(IPT)=C(NIPT)
140      NUMB=0
141      DO 355 I=1,20
142      355 FLAG(I)=0
143      WRITE(8)IC88,CARD88
144      GO TO 808
145      C
146      800 LINE=LINE+1
147      WRITE(OUT,440)LINE,TCT,XBACK
148      440 FORMAT(1X,I4,1X,I4,1X,24X,"TOTAL COST",F13.0,2X,"REASON",F12.0)
149      F12.0)
150      LINE=LINE+1
151      WRITE(OUT,18)LINE
152      ENDDO
153      DO 450 I=1,20
154      450 FLAG(I)=0
155      GO TO 1
156      C
157      C
158      850 STOP
159      END

```

2 01 10-20-79 10.000 28

1970 EQUALITY OR NON-EQUALITY COMPARISON MAY NOT BE NEARLY AS SOCIAL IF NOT

Subroutine: GETDAT**Functions**

This program reads output from GETBSO and retrieves corresponding LRU/SRU item description data from random file 08. It then outputs MOD-METRIC item description records with "IC2 = 1", and delivery schedule cards for later processing by the program EVALUATE.

Descriptions

This routine is used in conjunction with the program SAVDAT. Program SAVDAT saves item description data on a random file for subsequent retrieval by this program.

Program GETDAT begins by reading MOD-METRIC input card types 92, 96, 98, and 99 from random file 08, and immediately writes these cards to the output file 09. It then reads the first "97" card to determine the number of input records (NCARD) that is needed to hold the flying hour data. Next, the flying hour data cards are read.

The program then reads an LRU/SRU levels card from file 05, and decodes the sequence number that was associated with this LRU/SRU by the subroutine SAVDAT. This sequence number, ISEQ, is then used to retrieve the appropriate

item description data from random file 08. This information is then used to output a MOD-METRIC item description record with "IC2 = 1" to file 09. Next, the program writes a "95" delivery schedule card to file 05. These steps are then repeated for each LRU/SRU record on file 05.

The output from this routine serves as input to the program EVALUATE. The program EVALUATE reads the total number of assets recorded on the delivery schedule card, and determines an optimal allocation of these assets among the bases.

12 01 40023-79 17,304 78

```

03      DECODE(LECARD,03)XEND
04      22 FORMAT(10X)R2)
05      0
06      INDEXED(10) GO TO 070
07      DECODE(LECARD,04)LECARD4XEND0XEND
08      23 FORMAT(10X)R2)XEND
09      DECODE(LECARD,04)XEND
10      24 FORMAT(10X)R2)
11      0
12      READ(10X)LECARD
13      DECODE(LECARD,03)XEND
14      25 FORMAT(10X)R2)
15      0
16      26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000

```


2 08 40-23-79 17,284 58

05 2707
06 270

Subroutine: INTPRO

Function:

This program computes initial provisioning requirements according to logic specified in AFLCR 57-27.

Description:

This subroutine is called by the program DMSGN1 to compute initial provisioning requirements according to logic specified in AFLCR 57-27. The program was originally written by Mr. Terry Mitchell/AFALD/XRS as a CREATE Time-Sharing program. The program was subsequently modified for use in the Levels Computation System for use as a callable subroutine.

If the write flag IWT(3) equals 1, the routine prints out the details of its computations. The printed details are in the same form as an AFLCR 57-27 Recoverable Item Requirements Computation Worksheet, AFLC Form 614. An example of this form is presented in Figure II-2 of reference 1.

```

1      *NRUN=TIME/OBJ/R57-27.0(RCD,NOGO)
2      *R57-27.5
3      SUBROUTINE INTRO(WUC,OST,COST,MTBD,RNRTS,COND,BRCD,DRCD,
4      &                PLT,QPA,NSN,AVGMP,PEAKP,ITER)
5      CHARACTER WUC*8,NSN*15,CARD*70,DELSCH*42,WUCPN*24
6      INTEGER C1,C2,C2H
7      COMMON/INT/IWT(5)
8      C
9      IDBUG=IWT(3)
10     12 FORMAT(V)
11     IF(IDBUG,GE,1)WRITE(6,12) " ----INTRO--",
12     & " INITIAL PROVISIONING CALCULATION"
13     IF(IDBUG,EQ,1) WRITE(6,25) WUC,OST,COST,MTBD,RNRTS,
14     & COND,BRCD,DRCD,PLT,QPA,NSN,
15     & AVGMP,PEAKP
16     25 FORMAT("0","WUC=","A9," OST=","F5.1," COST=","F11.2,
17     & " MTBD=","F8.1," RNRTS=","F5.2," COND=","F5.2," BRCD=",
18     & "F5.1," DRCD=","F5.1," PLT=","F5.1," QPA=","F5.1,"/5X,
19     & "NSN=","A15,"/10X,"AVGMP=","F8.0," PEAKP=","F5.2//)
20     IF(IDBUG,EQ,1) WRITE(6,30) WUC,NSN
21     30 FORMAT("0",12X,"R57-27 COMP FOR WUC ","A9," NSN ","A15)
22     IQPA=QPA
23     IF(IDBUG,EQ,1) WRITE(6,40) IQPA,COST
24     40 FORMAT("0",17X,I5,3X,F10.2)
25     DRCM=AINT( (DRCD/30.) * 10. + 0.5 ) / 10.
26     IF(IDBUG,EQ,1) WRITE(6,50) PLT,BRCD,DRCD,DRCM,OST,AVGMP,PEAKP
27     50 FORMAT("0",10X,F5.0,F5.0,F5.0,F5.1,F5.0,F5.0,6X,F6.1///)
28     BPP=1.0-RNRTS
29     TOIM=AINT( (100./MTBD) * 10000. + 0.5 ) / 10000.
30     TOIMD=AINT( RNRTS * TOIM * 10000. + 0.5 ) / 10000.
31     TOIMB= TOIM-TOIMD
32     WOP=AINT( RNRTS * COND * 10000. + 0.5 ) / 10000.
33     WOR=AINT( TOIM * WOR * 10000. + 0.5 ) / 10000.
34     IF(IDBUG,EQ,1)
35     & WRITE(6,60) COND,BPP,RNRTS,TOIM,TOIMD,TOIMB,WOP,WOR
36     60 FORMAT("0",5X,F5.2,F5.2,F5.2,5F7.4///)
37     PCSL=1.0 * AVGMP * QPA
38     PCSLR=AINT( PCSL * WOR * 10. + 0.5 ) / 10.
39     PCSLD= PCSLR * COST
40     IF(IDBUG,EQ,1) WRITE(6,70) PCSL,WOR,PCSLR,PCSLD
41     70 FORMAT("0",25X,F15.4,F10.4,F10.4,F10.2)
42     RLTP= PLT * AVGMP * QPA
43     RLTPR= AINT( RLTP * WOR * 10. + 0.5 ) / 10.
44     RLTPD= RLTPR * COST
45     IF(IDBUG,EQ,1) WRITE(6,70) RLTP,WOR,RLTPR,RLTPD
46     DRCF= DRCM * AVGMP * QPA
47     DRCPR= AINT( DRCF * TOIMD * 10. + 0.5 ) / 10.
48     DRCPD= DRCPR * COST
49     IF(IDBUG,EQ,1) WRITE(6,70) DRCF,TOIMD,DRCPR,DRCPD
50     IF(IDBUG,EQ,1) WRITE(6,80)
51     80 FORMAT("0")
52     BRCP=AINT( (BRCD * PEAKP * QPA / 30. ) * 10000. + 0.5 ) / 10000.

```

INTRO

12 01 10-20-79 10.375 .5

```
53 BRCPR=AIMT( BRCR * TOIMB * 10. + 0.5 ) / 10.
54 BRCPD= BRCPR * COST
55 IF(IDBUG.EQ.1) WRITE(06,70) BRCR,TOIMB,BRCPR,BRCPD
56 BOSTP=AIMT( ( OST * PEAKP * QPA / 30. ) * 10000. + 0.5 ) / 10000.
57 BOSTPR=AIMT( BOSTP * TOIMD * 10. + 0.5 ) / 10.
58 BOSTPD= BOSTPR * COST
59 IF(IDBUG.EQ.1) WRITE(06,70) BOSTP,TOIMD,BOSTPR,BOSTPD
60 IF(IDBUG.EQ.1) WRITE(06,80)
61 BSLR= BRCPR + BOSTPR
62 BSID= BSLR * COST
63 IF(IDBUG.EQ.1) WRITE(06,90) BSLR,BSID
64 90 FORMAT("0",50X,F10.4,F10.2)
65 TRR=AIMT( PCSLR + RLTPR + BRCPR + BSLR + 0.5 )
66 TRD=TRR * COST
67 IF(IDBUG.EQ.1) WRITE(06,90) TRR,TRD
68 ITRR=TRR
69 RETURN
70 END
```

Subroutine: LEVLDP

Function:

This routine produces duplicates of levels records read in on file 07, and outputs the duplicates to file 08. The program is used whenever AFLCR 57-27 logic is used to compute initial provisioning levels.

Description:

Programs within the MOD-METRIC system produce a different set of stock levels for each given Buy Support Objective (BSO). In our study, five different BSOs were used, resulting in five sets of stock level records for each LRU and SRU being considered. However, a desired Buy Support Objective is not considered when AFLCR 57-27 logic is used. This program is employed to produce the same number of LRU/SRU stock level records for the AFLCR 57-27 computation logic as is produced whenever a MOD-METRIC computation routine is used.

The program begins by reading the "BSO-FILE" which contains the set of desired Buy Support Objectives to be evaluated in the current computational run. It reads this file and counts the number of BSO records on the file, NBSO. The routine then produces NBSO duplicates of each levels record read in from file 07. These duplicates are output to file 08. A counter in the part number identification field, representing the number of BSO being considered, is updated for each output record.

19T 01 10-20-79 15.518 JS

```

1  *SRUN=LINE/082/LEVLP.0(SCD)*LINE/06/BSOFILE*05*.R
2  *LEVLP.S
3  C
4  C      THIS ROUTINE FIRST READS THE BSO-FILE, AND COUNTS
5  C      THE NUMBER OF BSO-RECORDS; NBSO.
6  C
7  C      THE ROUTINE THEN PRODUCES DUPLICATES OF LEVELS
8  C      RECORDS READ IN ON FILE 07, AND OUTPUTS THE DUPLICATES
9  C      TO FILE 08. A COUNTER, REPRESENTING THE NLAN, IS UPDATED
10 C      ON EACH OUTPUT RECORD.
11 C
12 C      CHARACTER FIELD1*11, FIELD2*64
13 C
14 C      NBSO=0
15 5  READ(5,33,END=10)FIELD1
16  NBSO=NBSO+1
17  GO TO 5
18 C
19 C      READ IN AND PRODUCE NBSO DUPLICATES OF THE LEVELS RECORDS.
20 C
21 10 CONTINUE
22  WRITE(6,18)NBSO
23 13 FORMAT(///20X,"NO. OF BSO'S 4"X13,/)
24 C
25  REWIND 07
26 20 CONTINUE
27  READ(07,33,END=800)FIELD1,NLAN,FIELD2
28 33 FORMAT(11,2,A64)
29  IF(NLAN.LT.10)WRITE(6,35)FIELD1,NLAN,FIELD2
30 35 FORMAT(11,"09.11,A64)
31  IF(NLAN.GE.10)WRITE(6,33)FIELD1,NLAN,FIELD2
32 C
33 C      OUTPUT DUPLICATES
34 C
35  DO 60 NLAN=1,NBSO
36  IF(NLAN.GE.10)WRITE(6,33)FIELD1,NLAN,FIELD2
37  IF(NLAN.LT.10)WRITE(6,35)FIELD1,NLAN,FIELD2
38 60 CONTINUE
39 C
40  GO TO 20
41 C
42 C      END OF RUN
43 C
44 800 CONTINUE
45  REWIND 08
46  WRITE(6,803)
47 803 FORMAT(120,"END OF RUN")
48  STOP
49  END

```

LEVLP

Subroutine: LEVLPI

Function:

This program reads levels cards for quarter I, and outputs levels cards for quarters I and I+1.

Description:

In the Recoverable Item Management Evaluator, item stock levels are recomputed on a quarterly basis. Unfortunately, it was found that the MOD-METRIC computation system required very large amounts of computer time to determine stock levels for the item sets selected for consideration in this study. Consequently, to reduce compute time requirements to reasonable levels, it was decided to recompute stock levels on a six month basis, rather than quarterly. Program LEVLPI was developed to resolve interface problems associated with six months levels computations performed by the Levels Computation System, and the Quarterly Update calculations performed in the RIME simulation model.

The function of this program is to produce duplicates of input LRU/SRU stock level records. Specifically, the routine first reads a levels card for quarter I from file 07. It then writes out to file 08 a copy of this card, followed by an identical card with coding appropriate for quarter I+1. In RIME, levels are still recomputed quarterly; however, identical values are read for quarters 1 and 2, 3 and 4, and so on.

```
1  *#RUN=TIME/GRJ/LEVEL1.S(BCD)TIME/GO/LEVELS.C"07".B;
2  *#  TIME/HOLD"08".W
3  *LEVEL1.S
4  C
5  C      READ LEVELS CARD FOR QUARTER I.
6  C      OUTPUT LEVELS CARDS FOR QTR I AND I+1.
7  C
8  C      CHARACTER LEVELCD=77
9  C
10  IDBUG=1
11  CALL FPARAM(1,132)
12  IREC=0
13  C
14  REWIND 07
15  REWIND 08
16  C
17  10 CONTINUE
18  C
19  READ(7,23,END=800) LEVELCD
20  23 FORMAT(A77)
21  IREC=IREC+1
22  IF(IDBUG.GE.1)PRINT,IREC,LEVELCD
23  C
24  DECODE(LEVELCD,33)IQTR
25  33 FORMAT(T16,X2)
26  WRITE(8,23)LEVELCD
27  C
28  NQTR=IQTR+1
29  IF(NQTR.LT.10)ENCODE(LEVELCD,53)NQTR
30  IF(NQTR.GE.10)ENCODE(LEVELCD,33)NQTR
31  C
32  53 FORMAT(T16,"0",I4)
33  C
34  C      WRITE OUT LEVELS CARD FOR QTR I+1=NQTR
35  C
36  WRITE(8,23)LEVELCD
37  GO TO 10
38  C
39  800 CONTINUE
40  PRINT,"  LEVEL1--7--RECORDS PROCESSED=",IREC
41  STOP
42  END
```

LEVEL1

Subroutine: METINP**Function:**

This routine estimates Mean Time Between Demands (XMTBD), NRTS rates (XNRTS), and condemnation rates (CONS) using historical D041 histories.

Calling Parameters:

- IS = The quarter number identifying the most recent historical period to be used in rates estimation. (Note: Quarter 1 represents the oldest quarter, while quarter 16 represents the most recent quarter for which data is available).
- IE = Quarter number identifying the oldest historical period to be used for rates estimation.
- J = The item index identifying the particular LRU or SRU for which rates are to be developed. J = 1 denotes the LRU, J = 2 denotes the first SRU, J = 3 denotes the second SRU, and so on.

Description:

Subroutine METINP begins by computing the total base reparable generations (ITBRGN), total LRU program (ITPROG), the total number of NRTS events

(ITNRTS), and the total number of base condemnations (ITBCND) occurring during the quarters IS through IE, inclusive. It then computes values for Mean Time Between Demands (XMTBD(J)), the NRTS fraction (XNRTS(J)), and the base condemnation fraction (CONS(J)) for item number J. To prevent "fatal" computation errors in subsequent MOD-METRIC routines, each of these values are compared to upper and lower bounds, and set to the appropriate bounds if they are out of range. If XMTBD(J) exceeds 99999, it is reset to this value. Similarly, if XMTBD(J) is less than 1, it is also set to 99999. This latter case implies that there has been no program for item number J, and consequently the estimated MTBD is set to a very large number.

Similarly, if there have been no base generations for item number J during the interval IS through IE, the estimated NRTS fraction is set to $XNRTS(J) = 0.01$. If there have been at least some base rep gens in past periods, but no NRTS actions have been recorded, the estimated NRTS fraction is also set to 0.01. Hence, the estimated NRTS fraction is never allowed to be less than 0.01. This is because the MOD-METRIC program TWOIND will not accept as input any item with a NRTS fraction less than this value.

95T 01 10-20-79 13.870

```

1      SUBROUTINE METINP(I3,I2,J)
2      COMMON/IDBGN/IDBGN(16,40)
3      COMMON/ITPROG/ITPROG(16,40)
4      COMMON/INTSTS/INTSTS(16,40)
5      COMMON/ITCND/ITCND(16,40)
6      COMMON/XMTBD/XMTBD(40)
7      COMMON/XMRTS/XMRTS(40)
8      COMMON/CONS/CONS(40)
9      COMMON/IWT/IWT(20)
10     C
11     IDBUG=INT(4)
12     C      INITIALIZE INTERMEDIATE SUMS:
13     ITBGN=0
14     ITPROG=0
15     INTSTS=0
16     ITCND=0
17     C      COMPUTE INTERMEDIATE SUMS.
18     C      USE LRU PROGRAM IN XMTBD CALCULATIONS,
19     C      SINCE IT IS ASSUMED THAT SRU ARE PECULIAR
20     C      TO THE LRU. (THIS PREVENTS INCONSISTENCIES
21     C      THAT MIGHT CREEP IN IF THE PROGRAM DATA IS
22     C      INCONSISTENT.)
23     DO 100 I=1,I3
24     ITBGN=ITBGN+IDBGN(I,J)
25     ITPROG=ITPROG+ITPROG(I,J)
26     INTSTS=INTSTS+INTSTS(I,J)
27     ITCND=ITCND+ITCND(I,J)
28     100 CONTINUE
29     C      CONVERT ITPROG TO HOURS.
30     ITPROG=ITPROG*100
31     C
32     IF(IDBUG.GE.1)WRITE(6,103)I3,I2,ITPROG,ITBGN,
33     & INTSTS,ITCND
34     103 FORMAT('---METINP PROGRAM RESULTS FOR QTR ',I2,
35     & ' THRU QTR ',I2/ 'T20:',ITPROG,'T10:',
36     & 'ITBGN=',
37     & 'INTSTS=',I10,' ITCND=',I10)
38     C
39     C      COMPUTE MEAN TIME BETWEEN DEMANDS.
40     C
41     IF(ITBGN.GT.0) GO TO 150
42     XMTBD(J)=99999.
43     GO TO 175
44     150 CONTINUE
45     XMTBD(J)=FLOAT(ITPROG)/FLOAT(ITBGN)
46     IF(XMTBD(J).GT.99999.0) XMTBD(J)=99999.0
47     IF(XMTBD(J).LT. 1.) XMTBD(J)=99999.0
48     C      COMPUTE RATE RATE (NOT NEARLY THIS STATION.)
49     175 IF(ITBGN.GT.0) GO TO 200
50     XMRTS(J)=0.01
51     GO TO 210
52     200 CONTINUE
53     XMRTS(J)=FLOAT(INTSTS)/FLOAT(ITBGN)

```

METINP

ST 01 10-20-79 12.870

```

53      IF(XNRTS(J).LT.0.01) XNRTS(J)=0.01
54      C
55      C   COMPUTE CONDENSATION RATE (FUNCTION OF NRTS RATE)
56      210 IF(ITNRTS.GT.0) GO TO 300
57      CONS(J)=0.0
58      GO TO 310
59      300 CONTINUE
60      CONS(J)=FLOAT(ITBCND)/FLOAT(ITNRTS)
61
62      310 CONTINUE
63      IF(IDBUS.GE.1)WRITE(6,313)J,SHRBD(J),XNRTS(J),CONS(J)
64      313 FORMAT(1X,I2," ESTIMATES="," ",
65      &      "XNRTS=","F8.3," "XNRTS=","F8.3," "CONS=",
66      &      "F8.3)
67      C
68      RETURN
69      END

```

Subroutine: ONEIND

Function:

This MOD-METRIC program computes stock levels using assumptions embedded in Sherbrooke's original METRIC model.

Description:

This program computes optimal depot and base stock levels ignoring LRU/SRU relationships. The computational logic and input/output requirements for the program are defined in AFLCP 57-13, and will not be discussed further here.

The RIME version of this routine is identical with the original MOD-METRIC program with two exceptions. First, the input program has been modified to read values of the levels computation variables IMINSK, BOMIN, and JEQBAS from the "99" program parameter card. If JEQBAS = 1, the flying hour and order and ship time data by base is reset to the average values summed over all bases. These changes are implemented in program lines 1531-1545.

The second change involves implementation of upper and lower bounds on the stock levels computation. If IMINSK = 1, the expected number of assets in the repair/resupply pipeline is used as a lower bound on the computed stock level. Also, in this case, no additional stock is allocated to a base after the expected backorders at that base is reduced below BOMIN/NBASES. These stock level limitation calculations are implemented in program lines 2571-2605 and 2952-2963.

```

1 *SRUN=IRYNE/ONE/ONEIND.O(SCD.WOBY)
2 *ONEIND.S
3 C NAME=NET, VERSION 3
4 C COMPILED FILE IS MODNETIC/ONEIND.S
5 C
6 DIMENSION PH(23),OST(23),BRT(100),DET(100),DS(15,100),
7 KNO(100),CENNU(100),YMTW(100),YMTS(100),APP(100),
8 TBACK(20),BBACK(100,23,3),ECOST(3),WLRUD(100,3),ADSLT(23),
9 JSTCK(23),HRAWN(100),WLRUD(100,23,3),DELTA(1000),GN(25),
10 BAKO(20),BUDG(20),RATE(20),LENE(100),MONTHS(12),
11 CON(100),PLT(100),PO(100),YTST(100),PDSO(100)
12 & TPNOB(28),TPILL(23),FWWT(23),PNOB(100,3),FILL(100,3)
13 C
14 REAL LERT,LAGRAN,LAGR(2)
15 C
16 INTEGER ND6(17),ID(100,6),ZAC(100),NSN(100,4),IDNSN(6)
17 C
18 DATA KEY/6H267.AQ/
19 DATA NEN/4Q0*4H /
20 DATA LENS/1Q6*4H /,MARK/6H /,KBLNK/4H /
21 DATA MONTHS/9H JAN ,5H FEB ,5H MAR ,5H APR ,5H MAY ,
22 & 5H JUN ,5H JUL ,5H AUG ,5H SEP ,5H OCT ,5H NOV ,5H DEC /
23 C
24 CHARACTER CARD*72,NAME
25 C
26 DIMENSION IDORD(9)
27 C
28 I97=1
29 I98=1
30 I99=1
31 INDO=0
32 PRD=3
33 INOJ=0
34 NUNAB=100
35 MAXSTK=1000
36 CALL ISNT(I2NUNB,IDORD)
37 CALL DETH (I2NUNB)
38 5 READ(4,6,ERR=9,END=9)NAME
39 IF(NAME,9,"LAST ")GO TO 5
40 BACKSPACE 4
41 WRITE(6,7)IDATE,I2NUNB,IDORD
42 6 FORMAT(1X,A6)
43 7 FORMAT(1X,ONEIND ",1X,I6,1X,I5,1X,I9A6)
44 8 FORMAT(1X,"LAST ")
45 WRITE(6,8)
46 9 CALL DETCH(4,ISTAT, )
47 IYR=IDATE/10000
48 IKK=IYR*10000
49 IHO=(I2NUNB-IKK)/100
50 IKK=IKK+IHO*100
51 IDAY=I2NUNB-IKK
52 IHO=MONTHS(IHO)

```

ONEIND

T 01 10-20-79 18.389 JS

```

53      C
54      C      DETERMINE TYPE OF INPUT
55      1 IF(I99.EQ.1.AND.I98.EQ.1.AND.I97.EQ.1)GO TO 400
56      READ(5,2)IC1,IC2,CARD
57      2 FORMAT(2E1,1X,A72)
58      IF(IC1.EQ.9)GO TO 4
59      WRITE(6,3)I99,I98,I97
60      3 FORMAT(5H1I99=,I2,5X,4HI98=,I2,5X,4HI97=,I2,10X,5HERROR8/)
61      STOP
62      C
63      C      PROGRAM PARAMETERS-- 92,97,98,99
64      4 I9=0
65      GO TO(4,92,1,1,1,1,97,98,99),IC2
66      C
67      C      92-- HEADING/TITLE FOR TOP OF PAGE
68      92 DECODE(CARD,92)HDG
69      192 FORMAT(17A4)
70      GO TO 1
71      C
72      C      97-- FLYING HOURS & ORDER AND SHIP TIME PER BASE
73      C
74      97 I97=1
75      C
76      C      INPUT EQUAL BASE FLAG(YES/DAS) AND
77      C      NUMBER OF BASES(NBASSES).
78      DECODE(CARD,97) NBASSES
79      197 FORMAT(I2)
80      IF(NBASSES.GT.23)GO TO 897
81      IB=1
82      LB=MINO(6,NBASSES)
83      297 DECODE(CARD,97)IPH(I),OST(Y)YI=IB,LB)
84      397 FORMAT(13X,6(P6.0,P3.0))
85      IF(LB.EQ.NBASSES)GOTO 597
86      IB=IB+8
87      LB=MINO(LB+8,NBASSES)
88      READ(5,2)IC1,IC2,CARD
89      IF(IC1.EQ.9.AND.IC2.EQ.7)GO TO 297
90      496 WRITE(6,497)
91      497 FORMAT(28H0FLYING HOUR CARD INCOMPLETE)
92      STOP
93      597 FLYTOT=0
94      AOST=0
95      ADDRT = 0.0
96      OMIN = 1.E35
97      DO 697 K=1,NBASSES
98      AOST=AOST+OST(K)
99      OMIN = AMIN1 ( OST(K),OMIN )
100      697 FLYTOT=FLYTOT+PH(K)
101      DO 797 K=1,NBASSES
102      ADDRT = ADDRT + OST(K) * PH(K)
103      797 PHWT(K)=PH(K)/FLYTOT
104      ADDRT = ADDRT / FLYTOT = OMIN

```

T 01 10-20-79 10.989 3

```

57 512 FORMAT('0NOUT',I2,' IPHWT',I2,' IPWCH',I2,' IPSOW',I2)
58 WRITE(ROUT,515)
59 515 FORMAT(28H0BASE FLY HRS OST )
60 DO 520 K=1,NBASES
61 520 WRITE(ROUT,525)K,PH(K),OST(K)
62 525 FORMAT(1X,I4,4F9.0)
63 530 WRITE(ROUT,535)
64 535 FORMAT(//6X,12HWORK PART,10X,
65 & 43HALC UNIT MEAN TIME(HRS) REMOVALS PER,3X,
66 & 40HNRTE COND REPAIR TIME SECUR LEAD OPN,3X,
67 & 6HSYSTEM,/
68 & 6X,CUNIT NUMBER',"/NSH",4X,
69 & 43HCODE COST BETWEEN DEMANDS 100 FLY HRS,3X,
70 & 36HRATE RATE BASE DEPOT TIME(MONTHS) , " HA",4X,
71 & 8HPIPELINE//)
72 C
73 C-----BEGIN XITE CHANGES FOR JEQBAS
74 IF(JEQBAS,NE.1)GO TO 599
75 C
76 C SET PARAMETERS FOR EQUIL BASE CALCULATIONS
77 C
78 DO 598 N=1,NBASES
79 PHWT(N)=ASPH/FLYTOT
80 PH(K)=ASPH
81 OST(K)=EOST
82 598 CONTINUE
83 599 CONTINUE
84 C-----END OF JEQBAS CHANGES
85 NU=0
86 PIPCST=0
87 RATE(1)=0.
88 TOTBND=0
89 BAKO(1)=0.
90 BUDG(1)=0.
91 BSOLST=1.E36
92 CHAX=1
93 CHIN=1.E80
94 BUDINC=0
95 C
96 600 NU=NU+1
97 IF(INOJ,NE.0)GO TO 2580
98 615 READ(5,2,END=9999)IC1,IC2,CKRD
99 IF ( IC1,LE.5 ) GO TO 605
100 GO TO(11,605,605,605,605,605,605,615),IC1
101 C
102 9999 INOJ=1
103 IF(NU,NE.1)GO TO 2580
104 NU=NU-1
105 GO TO 1000
106 615 IF(IC2,NE.1)GO TO 605
107 I9=1
108 NU=NU-1

```


T 01 10-20-79 11.389 JS

```

09      GO TO 2000
10      C
11      C      11-- BASIC DATA
12      C
13      11 IF(IC2.NE.1)GO TO 800
14      DECODE(CRRD,111)(ID(U,I),I=1,6),CPRU(U),
15      & YMTBF(U),YNRTS(U),CON(U),NAPP,BRT(U),DRT(U),
16      & PLT(U),ALC(U),PC(U)
17      111 FORMAT(6A4,3X,F7.0,F6.0,F5.2,F3.2,I2,3F3.0,A1,AU)
*W 1282 INCOMPATIBLE V.D FIELD IN 'F' SPECIFICATION
*W 1222 THE CHARACTERS PRECEDING ABOVE FORMAT ERROR ARE ,F3.2.
18      DRT(U) = DRT(U) + ADDRT
19      YDUMMY=YNRTS(U)
20      CDUMMY=CON(U)
21      APP(U)=MAX0(NAPP,1)
22      R=100.*APP(U)/YMTBF(U)
23      IF(YNRTS(U).LE.0)GO TO 180
24      CON(U)=CON(U)*YNRTS(U)
25      PNU=CON(U)*PLT(U)*30.*CFAC/YNRTS(U)
26      DNU=PNU+(YNRTS(U)-CON(U))*DRT(U)/YNRTS(U)
27      GO TO 185
28      C
29      180 YNRTS(U)=CON(U)
30      PNU=600
31      DNU=600
32      185 PIPE=0
33      C
34      DO 211 K=1,NBASES
35      TT=(1.+YNRTS(U))*BRT(U)+YNRWS(U)*(DNU+OST(K))
36      PIPE=PIPE+TT*APP(U)*PH(K)/[YMTBF(U)*30.]
37      211 CONTINUE
38      PIPCST=PIPCST+PIPE*CPRU(U)
39      BAKO(1)=BAKO(1)+PIPE
40      C
41      C
42      700 WRITE(ROUT,705)U,(ID(U,I),I=1,6),ALC(U),CPRU(U),YMTBF(U),
43      & R,YDUMMY,CDUMMY,BRT(U),DRT(U),PLT(U),
44      & APP(U),PIPE
45      705 FORMAT(1X,I3,1X,6A4,1X,A1,F5.0,F4.2,1.9X,F5.5,4X,2F5.2,86?0U
46      & 2F7.0,8X,F3.0,3X,F6.1)
47      C
48      CMAX=AMAX1(CMAX,CPRU(U))
49      CHIN=AMIN1(CHIN,CPRU(U))
50      ICHIN=0
51      PLT(U)=PNU
52      DRT(U)=DNU
53      BUDINC=BUDINC+CPRU(U)
54      C
55      IF(U.EQ.NUMAX)GO TO 2000
56      GO TO 600
57      C
58      800 IF(IC2.NE.6)GO TO 605

```

T 01 10-20-79 11.989 3

```

59      J=NU+1
60      DECODE(CRD,805)IDNSN
61      805 FORMAT(6E4)
62      DO 806 I = 1,6
63      806 IF(IDNSN(I) NE. ID(J,I))GO TO 605
64      DECODE(CRD,810)(NSN(J,I),I=1E4)
65      810 FORMAT(25X,4E4)
66      WRITE(MOUT,815)(NSN(J,I),I=1,6)
67      815 FORMAT(15X,4E4)
68      GO TO 805
69      C
70      2000 IF(MU.EQ.0)GO TO 500
71      TOTBUD=PEPCST*BSTART
72      WRITE(MOUT,2002)EPCST,BSTART,TOTBUD
73      2002 FORMAT(20HOTAL PIPELINE      5,F12.17
74      &      20H      TIMES BSTART =,F15.8/20X,2(6H-4m---)/
75      &      20H STARTING BUDGET =,F12.8//)
76      WRITE(MOUT,2005)BAO(1)
77      2005 FORMAT(10X'EXPECTED BASE BACKBENERS WITH ZERO STOCK =',F9.8)
78      CAVG=2.*BUDINC/ELOAT(MU)
79      BUDINC=ABIN1(BUDINC*.5,PBINC*EPCST)
80      BUDINC=AMAX1(CAVG,BUDINC)
81      DO 2010 K=1,MU
82      AA=PRD*APP(J)/(YHTBF(J)*30.6)
83      UDD=0.
84      DO 2040 N=1,NBASES
85      2040 UDD=UDD+FH(K)*AA*YHRTS(J)
86      RDD=UDD*BETA1
87      PDD=PRD/(BETA*ORT(J)+BPRD)
88      BACKD=RDD*(1.0-PDD)/PDD
89      DEV=SQRT(BACKD/PDD)
90      IDEV=DEV
91      MNIGH=BACKD+1.5*DEV+1.5
92      MNIGH=MA20(MNIGH,12)
93      IF(MNIGH.LE.MAXSTK) GO TO 2080
94      WRITE(6,2050) J,BACKD,MAXSTK-5
95      2050 FORMAT(10X'DEPT PIPELINE FOR LRU #',I6,' #',F8.2,
96      &      'DEPT MAX STOCK SET TO ',I6)
97      MNIGH=MAXSTK
98      2060 CONTINUE
99      MLOW=BACKD-DEV
100     MLOW=MENO(MLOW,MAXSTK-IDEV)
101     MLOW=MEXO(MLOW,17)
102     C-----v-----w-----x-----y-----z-----RHH CHANGES
103     C
104     C      IF ZHINK=1, SET MIN DEPT STOCK LEVEL EQUAL TO
105     C      THE EXPECTED PIPELINE
106     IF(INISH.NE.1) GO TO 2007
107     IBACKD=BACKD+.5
108     MLOW=MENO(MLOW,IBACKD)
109     MNIGH=MEXO(MLOW,MNIGH)
110     2007 CONTINUE

```

BT 01 11-20-79 11.389 JS

```

311      C
312      CALL ADENSE(DD,RDD,BACKD,MLOW,MHIGH,DELTA)
313      C
314      C      IF INHMK=1, SET MHIGH TO FIRST LEVEL WITH
315      C      A DEPT DELAY FACTOR(DELTA)<= 0.0001
316      C
317      IF(INHMK.NE.1) GO TO 2069
318      DO 2066 I=MLOW,MHIGH
319      IF(DELTA(I).LT.0.0001) GO TO 2066
320      2064 CONTINUE
321      GO TO 2069
322      C      RESET MHIGH
323      C
324      2066 MHIGH=I
325      MHIGH=MAX0(MLOW+1,MHIGH)
326      2069 CONTINUE
327      C-----4-END LINE CHANGES
328      C
329      MLIH=MHIGH-MLOW
330      MRANGE(J)=MLIH
331      DO 2070 I=1,MLIH
332      2070 DS(I,J)=DELTA(MLOW+I-1)
333      2010 KRHO(J)=MLOW
334      EPS=1.E-30
335      KSAFE=0
336      NPTS=1
337      2020 LAGR(1)=-1.0/CHAX
338      C ENTER WITH INCREMENTED BUDGET.
339      2030 LAGR(2)=0.
340      TCOST(1)=0
341      TCOST(2)=0
342      DO 2180 NBIS=1,NBIS
343      LAGRAN=(LAGR(1)+LAGR(2))*0.5
344      GH(KBIS)=LAGRAN
345      TCOST(3)=0
346      DO 2150 J=1,NU
347      CLAGRA=LAGRAN*CBERU(J)
348      LCRIT=CLAGRA+1.
349      WHIN=9999.
350      RR=(1.+YHRTS(J))*BRT(J)
351      AA=PRD*APP(J)/(YHTBP(J)+30.0)
352      MLIH=MRANGE(J)
353      DO 2130 I=1,MLIH
354      W=0
355      DO 2110 K=1,NBASIS
356      TT=RR + YHRTS(J)*(OST(K)+DS(I,KHO,J)*BRT(J))
357      PE=BPRD/(BETA*TT+BPRD)
358      RE=FN(K)*AA*BETA1
359      R1=RE-1.
360      P1=1.-RE
361      RE=PE*RR
362      BACKN=RE*P1/PE

```

T 01 10-20-79 10.389 JS

```

63      NCUN=NE
64      STCKM=0
65      C-----LINE CHANGES
66      C          IF MINIMUM STOCK FLAG (IRINSK)=1, SET MINSTK
67      C          EQUAL TO THE EXPECTED PIPELINE
68      BBOLIN=BBOLIN/FLOAT(NBASES)
69      MINSTK=0
70      IF (IRINSK.EQ.1) MINSTK=BACKM+ 0.5
71      2090 CONTINUE
72      IF (STCKM.LT.MINSTK) GO TO 2098
73      IF (NCUN.GE.LCRITY) GO TO 2100
74      C
75      IF (BACKM.LT.BBOLIN) GO TO 2106
76      C
77      C-----END LINE CHANGES
78      C
79      IF (NE.STEPS) GO TO 2100
80      2093 CONTINUE
81      STCKM=STCKM+1.0
82      NE=(STCKM+1)*P1*NE/STCKM
83      BACKM=BACKM+NCUN-1.
84      NCUN=NCUN+NE
85      GO TO 2090
86      2100 CONTINUE
87      W=W+BACKM-CLAGRA*STCKM
88      JSTCK(N)=STCKM
89      TBACK(N)=BACKM
90      TPNOB(N)=NCUN
91      TPILL(N)=NCUN-NE
92      2110 CONTINUE
93      XENO=KENO(J)+IRNO-2
94      W=W-LAERAN*CPERU(J)*XENO
95      IF (W.GE.WMIN) GO TO 2130
96      WMIN=W
97      WLRUB(2,3)=XENO
98      PHOBO(2,2)=0.
99      FILL(J,3)=0.
100     DO 2120 N=1,NBASES
101     WLRUB(2,N,3)=JSTCK(K)
102     BBACK(2,N,3)=TBACK(K)
103     PHOBO(2,3)=PNOB(J,3)+TPNOB/KI*FWHT(K)
104     FILL(2,3)=FILL(J,3)+TPILL/KI*FWHT(K)
105     2120 CONTINUE
106     2130 CONTINUE
107     LSUM=KEND(J,3)
108     DO 2140 N=1,NBASES
109     2140 LSUM=LSUM+WLRUB(2,K,3)
110     TCOST(3)=TCOST(3)+FLOAT(LSUM)*CPERU(J)
111     2150 CONTINUE
112     IF (ICMEN.GT.0) GO TO 2155
113     IF (TCOST(1).ST.0) GO TO 2155
114     LAGR(1)=1/CHIN

```

BT 01 10-20-79 10.289 JS

```

415      ICHIN=1
416      GO TO 2080
417      2155 CONTINUE
418      L=1
419      IF(TCOST(3).GT.TOTBUD) L=2
420      LAGR(L)=LAGRAN
421      DO 2170 J=1,NU
422      WLRUD(J,L)=WLRUD(J,3)
423      PHOBO(J,L)=PHOBO(J,3)
424      FILL(J,L)=FILL(J,3)
425      DO 2160 N=1,NBASIS
426      WLRUN(J,N,L)=WLRUN(J,N,3)
427      2160 BBACK(J,N,L)=BBACK(J,N,3)
428      2170 CONTINUE
429      TCOST(L)=TCOST(3)
430      2180 CONTINUE
431      IF(IYRET.GT.3)WRITE(ROUT,2190) (CH(X),X=1,NBIS)
432      2190 FORMAT(13H0MULTIPLIERS=/1X,40B(2,3))
433      L=1
434      IF(TCOST(2).LE.0.)GOTO 2280
435      IF(TCOST(1).LE.0.) .OR. TCOST(1)=TOTBUD:LT,TOTBUD=TCOST(1) L=2
436      2200 WRITE(ROUT,2210)
437      C
438      2210 FORMAT(///,15(6X *****))
439      WRITE(ROUT,2220)
440      2220 FORMAT(5X,47HSUMMARY OF REQUIREMENTS, DISTRIBUTION AND COSTS)
441      WRITE(ROUT,2230)
442      2230 FORMAT(15(6X *****))
443      WRITE(ROUT,2240)
444      2240 FORMAT(6X,4HUNIT,4X,11HPART NUMBER,5X,3HALC,3X,4HNCOST,
445      & 3X,5HTOTAL,2X,5HDEPOT,2X,3HCON,JX,3HNBASIS)
446      BMIN=0.
447      CT=0.
448      DO 2245 J=1,NU
449      LTOT(J)=WLRUD(J,L)
450      DO 2246 N=1,NBASIS
451      BMIN=BMIN+BBACK(J,N,L)
452      2244 LTOT(J)=LTOT(J)+WLRUN(J,N,L)
453      2245 CT=CT+LTOT(J)*CPSRU(J)
454      IF(CT.LE.BDD8(NPTS))GOTO 2370
455      BSO=(BMIN-SAKO(NPTS))/(CT-BDD8(NPTS))
456      IF(BSO.LT.BSO1ST)GOTO 2370
457      C
458      2248 IF(INSO.NE.0)WRITE(ISO,2250)BSO,1D(1,1)1D(4,2).NU,BMIN,CT
459      C
460      2250 FORMAT(1X,3H88 ,E12,5,2A4,1X,E12,8,F12,07
461      DO 2280 J=1,NU
462      CH=FLOAT(LTOT(J))*CPSRU(J)
463      KC=(FLOAT(WLRUD(J,L))*999)/WRET(0)/DET(J)
464      KD=WLRUD(J,L)-KC
465      TTEND=0.
466      TTFIL=0.

```

T 01 10-20-79 10.569 CS

```

67      XMTBF=0.0
68      IF(IBSO.NE.0)WRITE(IBSO,2270)(ID(J,I),I=1,6),(NSN(J,I)*I*1X4),
69      &      ALC(J),CH,LTOT(J),KD,KC,
70      &      (NLRUB(J,K,L),K=1,NBASBSY
71      2270 FORMAT(1X,3H11,6A4,4A4,A1,F9.0,X4,2E3,25I3)
72      C
73      2280 WRITE(NOUT,2290)J,(ID(J,I),I=1,6),ALC(J),CH,LTOT(J),KD,KC,
74      &      (NLRUB(J,K,L),K=1,NBASBSY)
75      2290 FORMAT(1X,F3.2X,6A4,1X,A1,1X,F9.0,X7,2I6,23I3)
76      C
77      WRITE(NOUT,2300) CT
78      2300 FORMAT(//,38X,16H***** TOTAL COST,F12.0)
79      WRITE(NOUT,2310) BMIN
80      2310 FORMAT(38X,16H***** BACKORDERS,F12.6)
81      WRITE(NOUT,2320)BSO
82      2320 FORMAT(38X,17H***** BUY SUP OBJ,E12.5)
83      DO 4221 J=1,NU
84      XMTBF = XMTBF + 1./YMTBF(J)
85      TTPNB=BNSBO(J,L)*1./YMTBF(J) + TTPNB
86      TTFIL=FIAL(J,L)/YMTBF(J)+TTFIL
87      4221 CONTINUE
88      WRITE(NOUT,4226)TTPNB/XMTBF,TTFIL/XMTBF*100.
89      4226 FORMAT(38X,17H***** PROB O BAKK,F8.4/
90      &      38X,17H***** FILL RATE =F8.4)
91      IF(IERRT.GT.1)WRITE(NOUT,2321)
92      &      (J,PNSBO(J,L),FILL(J,L),J=1,NU)
93      2321 FORMAT(//OLNU PROBABILITY OF ENO BACKORDERS FILL RATE"/
94      &      10d(1X,I3,45X,F5.3,15X,F5.3A) )
95      IF(IERRT.LT.6)GOTO 2328
96      DO 2328 J=1,NU
97      JJ=NLRUB(J,L)-KNO(J)+2
98      2322 PDSO(J)=NSN(JJ,J)
99      WRITE(NOUT,2324)(J,PDSO(J),J=1,NU)
100      2324 FORMAT(/9 PROBABILITY OF BESO STOCKOUT FOR EACH ITEM"/
101      &      8(13(5X,F5.2)/))
102      2328 CONTINUE
103      C
104      IF(NPTS.EQ.2)GOTO 2330
105      IF((CT-BUDG(NPTS))/(BAKO(NPTS)-BMIN).LT.
106      &      (BUDG(NPTS)-BUDG(NPTS-1))/(BAKO(NPTS-1)-BAKO(NPTS)))GOTO 2340
107      2330 NPTS=NPTS+1
108      KSAHB=0
109      2340 BUDG(NPTS)=CT
110      BAKO(NPTS)=BMIN
111      RATE(NPTS)=TTFIL/XMTBF
112      BSOLST=BSO
113      CALL PTIME(ITIME)
114      IF(IERRT.GT.1)WRITE(NOUT,2350)ITIME
115      2350 FORMAT(//OLNED COMPUTE TIME",X9,1X HILLISECS.)
116      IF(NPTS.EQ.5)GOTO 2360
117      IF(LAGE(L).EQ.BESTOP)GO TO 2390
118      IF(NPTS.EQ.2)GOTO 2390

```

```

519      C
520      2360 TOTBUD=TOTBUD+BUDINC*FLOAT(NBNSZ)
521          LAGR(1)=MIN1(LAGR(1),LAGR(2))
522          GOTO 2030
523      2370 KSAHE=KSAHE+1
524          BUDINC=BUDINC+BUBINC
525          WRITE(NOUT,2190)(GM(I),I=1,NBS)
526          IF(KSAHE.LT.5)GOTO 2360
527          WRITE(NOUT,2180)
528      2380 FORMAT(31HNO PROGRAM LOOPING WITH SAME COST )
529      C-----+-----RIME CHANGE
530          IF(NPTS.GE.2) GO TO 2390
531      C-----+-----END CHANGE
532          GOTO 500
533      C
534      C
535      2390 WRITE(NOUT,2400)(NDG(I),I=1,16),BAKO(2),BAKO(2),BUDG(2)
536          & ,RATE(2)
537      2400 FORMAT(1H1,14X,29HPLOT OF BACKORDERS VS. BUDGET,5X,16A4/
538          & 11H BACKORDERS,72X,10HBACKORDERS,5X,6HBUDGET,2X,6HFLLRATE/
539          & 1X,F10.4,3H -X,69X,88.4,F13.0,2X,F8.4)
540          IF(IPNCH.EQ.0)GO TO 2430
541      C-----+-----RIME CHANGE-----STATEMENTS MOVED HERE FROM
542      C FOLLOWING STATEMENT 2005
543          IF(IPNCH.NE.0)WRITE(IPNCH,2420)(ID(1,I),ID(1,2),BUDG(1),BAKO(1)
544          & ,RATE(1))
545      C-----+-----END CHANGE
546          DO 2410 I=2,NPTS
547      2410 WRITE(IPNCH,2420)ID(1,1),ID(1,2),BUDG(I),BAKO(I),RATE(3)
548      2420 FORMAT(1X,A4,A2,F11.0,2F10.6)
549      2430 LAST=1
550          NP=2
551          DL=(BAKO(2)-BAKO(NPTS))/49.0
552          DK=(BUDG(NPTS)-BUDG(2))/99.0
553          DO 2520 I=3,NPTS
554          NEXT=(BAKO(2)-BAKO(I))/DL + 0.5
555          NB=NEXT-LAST+1
556          LAST=NEXT
557          IF(NB.LT.1)GOTO 2470
558          IF(NB.GT.70)GOTO 2520
559          DO 2460 J=1,NB
560          WRITE(NOUT,2440)
561      2440 FORMAT(2X,1H=)
562          IF(NP.GE.NPTS)GOTO 2460
563          NP=NP+1
564          WRITE(NOUT,2450)BAKO(NP),BUDG(NP),RATE(NP)
565      2450 FORMAT(1H+,82X,F8.4,F13.0,2X,N8.0)
566      2460 CONTINUE
567      2470 NK=(BUDG(I)-BUDG(2))/DK + 1.5
568          LINE(NK)=MARK
569          IF(NB.LT.0)GOTO 2490
570          WRITE(NOUT,2480)BAKO(I),LINE

```

8 01 10-20-79 10.989 .5

```

71 2480 FORMAT(1X,F10.4,2X=,100A1,1H)
72 LINE(NR)=KBLNK
73 GOTO 2510
74 2490 WRITE(ROUT,2490) LINE
75 2500 FORMAT(1X=.12X,100A1)
76 LINE(NR)=KBLNK
77 2510 IF(NR.NE.NPTS)GOTO 2520
78 NR=NR+1
79 WRITE(ROUT,2450)BAKO(NR),SUBGZWPY,NR,NR
80 2520 CONTINUE
81 WRITE(ROUT,2430)
82 2530 FORMAT(1X.10(10H.....IT/32X,1NX,5(19X,1NX))
83 B2=DK*80.0 + BUDG(2)
84 B3=DK*60.0 + BUDG(2)
85 B4=DK*80.0 + BUDG(2)
86 B5=DK*60.0 + BUDG(2)
87 WRITE(ROUT,2440)BUDG(2),B2,B3,B4,B5,BUDG(NPTS)
88 2540 FORMAT(7H BUDGET,F10.0,5F20.0)
89 WRITE(ROUT,2450)DL,DK
90 2550 FORMAT(12HOLINE VALUE=F7.5Y50.13HCOLUMN VALUE=F8.1)
91 IF(1PNCN.NE.0)WRITE(1PNCN,2560)
92 2560 FORMAT(1X1.48X)
93 GO TO 500

```

C

```

95 2580 WRITE(ROUT,2490)
96 2590 FORMAT(12HEND OF DATA)
97 STOP
98 END

```

*** 1470 EQUALITY OR NON-EQUALITY COMPARISON MAY NOT BE MEANINGFUL IN LOGICAL IF EX
 *** 7 MEMORY EXPANDED. USE \$LIMITS OR \$CORE= OPTION FOR NEXT RUN

Subroutine: OVHSTL

Function:

This routine computes overhaul stock levels using a 14-day supply rule.

Calling Parameters:

| | | |
|--------|---|---|
| IQTR | = | The quarter number for which stock levels are to be computed. |
| JJ | = | The item number for which stock levels are to be computed. |
| IOVSTL | = | The computed overhaul stock level. |

Description:

This routine first computes the total number of depot reparable generations for item number JJ which will be generated during the 4 quarter interval beginning with quarter IQTR. It then sets the computed overhaul stock level IOVSTL equal to a 14-day supply based upon the computed demand rate.

01 10-20-79 13.371

```

1      SUBROUTINE OVHSTL(IQTR,JJ,XOVSTL)
2      COMMON/IDRGW/IDRGW(16,40)
3      COMMON/TW2/TW2(20)
4      C
5      IDBUG=INT(6)
6      C
7      C      COMPUTE OVERHAUL STOCK LEVEL.
8      C
9      TDD=0.0
10     IS=IQTR
11     IF(IS.GT.13) IS=13
12     IE=IS+3
13     DO 100 I=IS,IE
14     TDD=TDD+IDRGW(I,JJ)
15     100 CONTINUE
16     C-----OVOST IS SET TO 15.0 -----
17     OVOST=15.0
18     XOVSTL=(TDD/365.0)*OVOST +0.5
19     C
20     IF(IDBUG.GE.1)WRITE(6,103)JJ,IQTR,IS,IE,TDD,OVOST,
21     XOVSTL
22     103 FORMATT/5---OVHSTL-ITER=Y,N2,"  QTR=Y,I3,"  IS=Y,I3,
23     IE=Y,I3,"  TDD=Y,F10.08
24     "  OVOST=Y,25.0,"  XOVSTL=Y,I5)
25     C
26     RETURN
27     END

```

OVHSTL

Subroutine: SAVDAT

Function:

This program saves item description data on random file 08. This information is subsequently retrieved by program GETDAT using a sequence number assigned by program SAVDAT.

Description:

This program is called prior to a ONEIND levels computation.

The program reads MOD-METRIC input records from file 05, and assigns a sequence number ISEQ to each incoming data card. It then writes each input card to the random file 08 using ISEQ as an index for the random file. The index ISEQ is then inserted in to the part number field of the input card, and the modified input record is written as output to file 09. This process continues until all records in the input file had been read, stored, and modified.

Outputs to file 09 are subsequently used as input to the ONEIND levels computation program. Later in the job processing stream, the sequence number ISEQ is used to retrieve information from the random file 08, and to relate it to computed stock levels. For additional discussions of this process, see the description of program GETDAT.

01 10-20-79 14.896 JS

```

1  *BRUN=ARINS/002/SAVDAT.0(BCD,NO007
2  *SAVDAT,8
3  C      THIS PROGRAM SAVE ICD=1 DATA ON RANDOM FILE "00".
4  C      THIS INFORMATION IS THEN RETRIEVED BY PROGRAM
5  C      GETSAT.8 FOLLOWING GETSSO. THE SEQUENCE NUMBER
6  C      "ISSO" IS USED AS ONE DATA RETRIEVAL KEY.
7  C
8  C
9  C      CHARACTER ICARD*72,ICARD1*10,ICARD2*50
10 C
11 C      SET RANDOM FILE RECORD SIZE(SEE P. 6-33 OF FORTRAN
12 C      MANUAL FOR DETAILS)
13 C
14 C      CALL NANSZ/00,127
15 C
16 C      IREC=0
17 C
18 10 READ(5,23,END=900)ICARD
19 IREC=IREC+1
20 23 FORMAT(57H)
21 C
22 C      DECODE(ICRD,23)IC1,IC2,ICRD1,ICARD2
23 23 FORMAT(2I1,1X:A10,3X,A50)
24 C
25 C      RECORD ICARD2 DATA ON FILE "00" FOR RECALL AFTER
26 C      GETSSO ROUTINE
27 C
28 C      ISSO=ISSO+1
29 C      WRITE(6,1220)ICARD
30 C
31 IF(IC2.NE.1) GO TO 50
32 IF(IC1.NE.9) GO TO 50
33 C
34 C      INSERT SEQUENCE NUMBER IN PART NUMBER FIELD.
35 C      AND OUTPUT THE RECORD. THIS NUMBER WILL BE
36 C      USED TO RETRIEVE ONE ICARD2 DATA AFTER
37 C      GETSSO.
38 C
39 C      WRITE(9,43)IC1,IC2,ICARD1,ISSO,ICARD2
40 43 FORMAT(2I1,1X:A10,1I3,A50)
41 GO TO 40
42 C
43 C      OUTPUT RECORD AS READ
44 C
45 50 CONTINUE
46 WRITE(9,23)ICARD
47 C
48 60 CONTINUE
49 GO TO 40
50 C
51 C      END OF FILE REACHED, STOP PROGRAM
52 C

```

SAVDAT

01 10-20-79 14.598 28

3 900 CONTINUE
4 WRITE(6,903) "-=+SAVDAT=- END OF FILE REACHED AFTER",
5 4 " PROCESSING ".INED." INPUT CARDS"
6 903 FORMAT(I
7 STOP
8 END

P 01 10-20-79 10.296 28

EXT DATE 08-16-78 188 3/1

ELAPSED
TIME (SECT)LINES/
MINUTE

| | | |
|--------|-----|-------|
| REHEAD | .18 | |
| ISS 1 | .09 | 36766 |
| ISS 2 | .00 | |
| ISS 4 | .08 | 39871 |
| ISS 5 | .30 | 11267 |
| TOTAL | .67 | 5126 |

AL TIME .70

THE WERE NO DIAGNOSTICS IN ABOVE COMPILEATION
5K WORDS WERE USED FOR THIS COMPILEATION

Subroutine: SPNDMS

Function:

This TimeSharing-Program is used to generate the Job Control Language statements required to utilize the Recoverable Item Management Evaluator System.

Description:

This program is used to simplify the preparation of run request for the RIME system. For a detailed discussion of the functions of this routine, see the User Instructions Chapter of Volume I.

**RUN=;SPNDMS.0(ASCII)

*SPNDMS.S

DIMENSION ISTAT(2)

CHARACTER CARD*72

CHARACTER SIMCAR*50

CHARACTER ANS*1

DIMENSION RP(10,13)

DATA RP/1.,0.,0.,1.,0.,0.,0.,0.,0.,0.,

3.,1.,0.,3.,1.,0., 0.,0.,0.,0.,

1.,0.,1.,1.,0.,1., 0.,0.,0.,0.,

1.,0.,0.,1.,0.,0., 1.,0.001,1.,0.001,

3.,1.,1.,3.,1.,1., 1.,0.001,1.,0.001,

2.,0.,0.,2.,0.,0., 0.,0.,0.,0.,

2.,0.,0.,2.,0.,0., 1.,0.01,1.,0.01,

4.,0.,0.,4.,0.,0., 0.,0.,1.,0.001,

2.,0.,0.,3.,1.,1., 0.,0.,1.,0.001,

2.,0.,0.,3.,1.,1., 1.,0.01,1.,0.001,

4.,0.,0.,1.,0.,0., 0.,0.,1.,0.001,

2.,0.,0.,1.,0.,0., 0.,0.,1.,0.001,

2.,0.,0.,2.,0.,0., 1.,0.01,1.,0.001/

RP(J,I)=J-TH PARAMETER FOR INVENTORY POLICY I.

1 CONTINUE

SIMCAR=";TAPE:07,X9D,,70053,,EXOGF"

READ OUTPUT CONTROLS

PRINT,"AC? PUNCH? SIMULATE? MORE CORE? (1=YES, 0=NO)"

IMICRO=0

READ,IAC,IPNCH,ISIMU,ICORE

5 IF(ISIMU.NE.1) GO TO 10

PRINT," IS --",SIMCAR

PRINT," OK FOR EXOGFILE? (Y=YES)"

READ,ANS

IF(ANS.EQ."Y")GO TO 10

PRINT," INPUT EXOGFILE CONTROL CARD"

READ(5,3)SIMCAR

3 FORMAT(A50)

PRINT," CARD WAS READ AS--"

PRINT,"====",SIMCAR

PRINT," IS CARD OK?(Y OR N)"

READ,ANS

IF(ANS.NE."Y") GO TO 5

10 CONTINUE

READ SIMULATION PARAMETERS

16 CONTINUE

PRINT 13," NFGRP, NLGRP,INQTR,NDHIS, CPU-LIMIT"

READ(5,13) NFGRP, NLGRP,INQTR,NDHIS,ICPULM

IF(ICPULM.GE.10) GO TO 17

PRINT," CPU-LIMIT MUST BE =>10"

GO TO 16

17 CONTINUE

ITOT=NDHIS+INQTR

IF(ITOT.LE.16) GO TO 19

SPNDMS

67

```

19 CONTINUE
C-----CREATE A TEMPORARY FILE 09-----
C
30 CONTINUE
  CALL CREATE(9,1,0,ISTAT(1),)
  IF(ISTAT(1).NE.0)GO TO 9000
C
  PRINT 11
11 FORMAT(///)
  read ident of run desired
C
  PRINT 13,"IDENT?"
  READ(5,13)I
  set run parameters
C
  IDENT=I
  IMETH=RP(1,I)
  IEQBAS=RP(2,I)
  ICOST=RP(3,I)
  KMETH=RP(4,I)
  KEQBAS=RP(5,I)
  KCOST=RP(6,I)
  IMINSKI=RP(7,I)
  BOMINI=RP(8,I)
  KMINSK=RP(9,I)
  BOMINK=RP(10,I)
C
  READ MANAGEMENT METHOD CODES
C
  PRINT 13,"  IDENT IMETH IEQBAS ICOST KMETH KEQBAS KCOST"
  PRINT 14, IDENT, IMETH,IEQBAS,ICOST,KMETH,KEQBAS,KCOST
  13 FORMAT(V)
  14 FORMAT(I7,6I7)
C
  PRINT 13,"  IMINSK BOMINI KMINSK BOMINK"
  PRINT 15,IMINSK,BOMINI,KMINSK,BOMINK
  15 FORMAT(I7, F10.3,I7,F10.3)
C
C-----BUILD BATCH CARD DECK-----
C
  IF(IAC.EQ.1) WRITE(9,1013)
20 1013 FORMAT("010##NORM,R(AC)")
30 IF(IAC.NE.1) WRITE(9,1014)
40 1014 FORMAT("010##NORM")
50 WRITE(9,1023) IDENT,NFGRP,NLGRP,IMETH,IEQBAS,ICOST,
50% KMETH,KEQBAS,KCOST
70% ,IMINSK,BOMINI,KMINSK,BOMINK
30 1023 FORMAT("011$:IDENT:WP1462,XRS/DEMMY-----",
70% I2,"***",I2,"*",I2,"*","---",6I2/
00% "012$:NOTE:-----",2(I2,"-",F6.4,"=="))
10 IF(ICPULM.LT.100)WRITE(9,1026)ICPULM,ICPULM
20 IF(ICPULM.GE.100)WRITE(9,1027)ICPULM,ICPULM
30 1026 FORMAT(
40% "018$:LIMITS:",I2,"",,9K"/
50% "020$:PARAM:",I2/
60% "023$:SELECTA:RIME/GO/DMSGN.E1")
70 1027 FORMAT(
80% "018$:LIMITS:",I2,"",,9K"/

```

023\$:SELECTA:RIME/GO/DMSGN.E1)

WRITE DMSGN1 PARAMETERS TO FILE 09

68

WRITE(9,33)IDENT, IMETH,IEQBAS,ICOST,KMETH,KEQBAS,KCOST

33 FORMAT('025\$',8I4)

WRITE(9,34)IMINSK,BOMINI,KMINSK,BOMINK

34 FORMAT('026\$',2(I5,F10.5))

WRITE(9,23)NFGRP, NLGRP,INQTR,NDHIS

23 FORMAT('027\$',8I5)

-----OBTAIN MOD-METRIC PARAMETERS FROM RIME/DMS.CTL

WRITE(9,37)

37 FORMAT('030\$:SELECTA:RIME/DMS.CTL')

OBTAIN FLYING PROGRAMS FROM RIME/F111FH.D

IF(NFGRP.LE.13)WRITE(9,38)

38 FORMAT('032\$:SELECTA:RIME/F111FH.D')

IF(NFGRP.GT.13)WRITE(9,39)

39 FORMAT('032\$:SELECTA:RIME/F15FH.D')

-----GENERATE REMAINDER OF JCL

WRITE(9,1008)

1008 FORMAT('035\$:SELECTA:RIME/GO/DMSGN.E2')

WRITE(9,1018) IMETH

1018 FORMAT('040\$:NOTE:***** INITIAL PROV--IMETH=',I2)

WRITE(9,1028) IMETH

1028 FORMAT('050\$:SELECTA:RIME/GO/INPRV.E',I1)

WRITE(9,1038) KMETH

1038 FORMAT('060\$:NOTE:***** REPLENISHMENT--KMETH=',I2)

WRITE(9,1043)KMETH

1043 FORMAT('070\$:SELECTA:RIME/GO/REPLN.E',I1)

WRITE(9,1053)

1053 FORMAT('080\$:NOTE:*****SORT LEVELS')

WRITE(9,1063)

1063 FORMAT('090\$:SELECTA:RIME/GO/SORTL.E1')

IF(IPNCH.EQ.1) WRITE(9,1213)

1213 FORMAT('1200\$:NOTE:*****PUNCH LEVELS',

8/, '1210\$:SELECTA:RIME/GO/PUNCH.E1')

IF(ISIMU.NE.1)GO TO 1400

IF(ICORE.EQ.0) WRITE(9,1313)SIMCAR

1313 FORMAT('1300\$:NOTE:*****RUN RIMSIM SIMULATION'/

'1310\$:SELECTA:RIME/GO/RISMS.E2'/

'1320',A50)

IF(ICORE.NE.0)WRITE(9,1314)SIMCAR

1314 FORMAT('1300\$:NOTE:*****RUN RIMSIM BIG MODEL'/

'1310\$:SELECTA:RIME/GO/RISMS.E4'/

'1320',A50)

1400 CONTINUE

IF(IMICRO.EQ.1)WRITE(9,1414)

1414 FORMAT('1400\$:NOTE:*****COPY P1 FOR MICROFICHE'/

8'1410\$:SELECTA:RIME/GO/MICRO.E1')

WRITE(9,9003)

9003 FORMAT('100\$:ENDJOB')

-----SPAWN THE BATCH COMPILE-----

ENDFILE 09

```

) REWIND 09
) CALL CALLSS("RUN 09#","CDIN")
)C
)C-----DETACH THE TEMPORARY FILE-----
)C
) CALL DETACH(09,ISTAT,)
)C
)C-----CONTINUE ? ? ?
)C
) PRINT 13,"CONTINUE?(Y OR N?)"
) READ(5,13)ANS
) IF(ANS.EQ."Y") GO TO 30
)C
) PRINT," DO ANOTHER GROUP?(Y OR N)"
) READ(5,13)ANS
) IF(ANS.EQ."Y")GO TO 1
)C
) STOP
)C
)C
) 9000 CONTINUE
) PRINT,"THIS PROGRAM USES FILE 09 AS A TEMPORARY FILE, BUT"
) PRINT,"FILE 09 ALREADY EXISTS."
) PRINT," DO A --REMO CLEARFILES--"
) STOP
) END

```

Subroutine: TWOIND

Function:

The program TWOIND implements the MOD-METRIC two-indenture stock level computation model.

Description:

This program implements the MOD-METRIC two-indenture stock level computation model. Input requirements and computation logic for this routine is described in AFLCP 57-13, and will not be discussed further here. The RIME version of this routine is identical with the AFLCP 57-13 version with two exceptions. First, subroutine INPUT has been modified to read in values for the stock level computation variables IMINSK and BOMIN from the "99" input record. The value of the MOD-METRIC variable BSTOP is then set to the input value for BOMIN whenever the bounds flag IMINSK equals 1. Otherwise, no change to BSTOP is made. These changes are implemented in program lines 3061-3102.

The second RIME modification is implemented in subroutine OUTPUT. In this routine, if the bounds flag IMINSK equals 1, the stock levels computed by TWOIND are compared to the expected number of assets in the repair/resupply pipeline. If a stock level computed by TWOIND is below this value, the corresponding stock level is reset to the expected number of assets in the pipeline. On the other hand, if IMINSK is not equal to 1, no bounds calculations are performed.

No other changes to the program TWOIND were made.

F 02 10626-79 1A,250

```

13      INTEGER QSL(8),QSS(8524), LSP(8),SPF(48,8)
14
15      REAL CERR2B,LCRIT,LAGNTH,LAGMAX
16
17      DATA XFF,INDB,REDS,INWTRU,ISUCRU /870/
18      DATA NODT,ISNT,INQNT,IP1LW / 6, 870 /
19      DATA NTA,RETA1,BRBD,BSOLD,BSKOP,CYAS,PEINC
20      / 3.0 1.011111, 23.0 1.0, 0.0, 1.0, 0.05 /
21      DATA WRO,MAXTRU / 30,285 /
22      DIMENSION OPTEST(5),TESTIN(5)
23      DATA OPTEST / 6E09999% UNKNCON, UNKNBSE, UNKNOCIN, UNKNSTOCK /
24      LOGICAL OPTION(8)
25      EQUIVALENCE ( OPTION(4),ROUNDO )
26      DATA BLANKS / 6E /
27      DATA NITLW / 0 /, INOURS / 100 /
28
29      *****
30
31      DETERMINE TYPE OF INPUT
32      IF ( NSTRUS.GT.0 ) GO TO 3
33
34      1 IF ( XFF.GT.0 ) GO TO 1090
35      READ (5,2)INDB=999,ERR40401IC1,IC2,CARD
36      2 FORMAT(2I4)139A77)
37      IF ( NITLW.LE.0 ) GO TO 3
38      IF ( IC1.LE.0 .OR. ERR40401 ) GO TO 1
39      IF ( IC1.GT.0 .AND. INDB.LE.0 ) GO TO 1
40      3 GO TO 140230,304025070076.00701,IC1
41
42      PROGRAM BLANKTEST= 92.98798699
43
44      90 IF ( INWTRU.GE.1 ) GO TO 1000
45      IF (IC2.GT.0)GO TO 904
46      904- PROGRAM CONTROL OPTIONS CARD
47
48      DECODE(CARD,929) ( TESTIN(1),IOPT=1,NOPT )
49      929 FORMAT(12I8,1X)
50      DO 930 IOPT = 1,NOPT
51      930 OPTION(IOPT) = ,TABLE:
52      KOPT = 0
53      INNOPT = 0
54      DO 930 IOPT = 1,NOPT
55      IF ( TESTIN(IOPT)=BLANKS ) 931,933,931
56
57      931 DO 933 JOPT = 1,NOPT
58      IF ( TESTIN(IOPT)=OPTTEST(JOPT) ) 933,932,933
59      932 OPTION(JOPT) = ,TABLE:
60      WRITE(6,940) OPTTEST(JOPT)
61      KOPT = JOPT + 1
62      GO TO 934
63      933 CONTINUE
64
65      IF ( ZEROPT.LE.0.0 ) WRITE(6,940)

```

02 10426-79 18,450

```

5      IERRPT = IERRPT + 1
6      WRITE(8,948) TESTIN(IERRPT)
7      934 CONTINUE
8
9      C
10     935 IF ( IERRPT * IERRPT ) 936-936-938
11     936 WRITE(8,937)
12     937 FORMAT(" *****/" "***** BLANK PROGRAM CONTROL CARD".
13     " * ENCOUNTERED"/" ***** STANDARD OPTIONS SET"/
14     " *****//")
15     GO TO 942
16     938 WRITE(8,940)
17     IF ( IERRPT ) HERR = HERR
18     IF ( IERRPT ) IERR = IERR
19     IF ( IERRPT ) IERR = 1000
20     IERR = 0
21     DO 939 IERR = 1, IERR
22     IF ( .NOT. OPTION(IERR) ) GO TO 939
23     WRITE(8,940) OPTION(IERR)
24     IERR = IERR + 1
25     TESTIN(IERR) = OPTIN(IERR)
26     939 CONTINUE
27     940 FORMAT(8H0*****/8H INVALID PROGRAM/8H CONTROL/
28     " 8H OPTIONS/")
29     941 FORMAT(5XA0)
30     942 FORMAT(8H1PROGRAM/8H CONTROL/8H OPTIONS/)
31     GO TO 1
32     901 GO TO (91,92,93,94,95,96,97,98,99),IC2
33
34     C
35     C      91-- DATA FILE COMMENT CARD
36     C
37     91 GO TO 1
38
39     C
40     C      92-- HEADING/TITLE FOR TOP OF PAGE#IC2OFIC2
41     C
42     92 DECODE(CARD,192)HDS
43     192 FORMAT(A59)
44     GO TO 1
45
46     C
47     93 DECODE(CARD,93339) TARGET
48     93339 FORMAT(F12.0)
49     94 GO TO 1
50     95 GO TO 1
51     96 GO TO 1
52
53     C
54     C      97-- FLYING HOURS AND UNDER/SHIP TIME FOR BASE
55     C
56     97 I97=1
57     DECODE(CARD,197)HBASES
58     197 FORMAT(I3)
59     IF (HBASES.GT.30) GO TO 1070
60     C
61     DO 975 IDS = 1, HBASES

```

```

57 975 IRRRT(IRS) = IRS
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
10
```


2 01 10626-79 181350

```

59 YNATE=THOURS*APP/YMTSD
60 WRITE (NOUT,421)HDS,XDAY,IMO,ITY
61 421 FORMAT(1H11A49/50X,12HDATE OF RUN ,13,A5,3H19;12)
62 WRITE(NOUT,521)HDS,HMTH,ESTOP,HBASES,CYAC,PRNC,IMJNSK,BSDMIN
63 521 FORMAT(6H0515=,13T5X,6HDATE,15,3,5X,
64 6HSTOP,15,2/ 7H 040930;13T5X,6HCFAC=,15,3,
65 6 2X,6HBTNC=,15,3, 1HJNSK =",16," BSDIN =",16,4)
66 WRITE(NOUT,324)HOUT,XPNT,IPNC,XFILE
67 324 FORMAT(4HOUT=" ,12T IPNT=" ,12, " IPNC=" ,12, " ISB=" ,12)
68 IF(CYAC;LE,0)WRITE(NOUT,921)
69 921 FORMAT(155X,750 CONSUMPTION STOCK COMPUTED"/)
70 WRITE(NOUT,621) IOST
71 621 FORMAT(15H05ASE FLS NBS ,4X,A5)
72 DO 721 ISB=1,HBASES
73 WRITE(NOUT,821) ISB,PR(INB),OUT(INB)
74 IF ( XBASEX ) WRITE(NOUT,722) XNBS(INB)
75 722 FORMAT(1H+126X,13)
76 721 CONTINUE
77 821 FORMAT(1X,14,2F9,0)
78 WRITE(NOUT,622) FLTFOF
79 622 FORMAT(8X,7H7--77-- / OH TOTAL , 18,0)
80 WRITE(NOUT,221) MTSD,THOURS
81 221 FORMAT(1X,16HWORK PART NO.,6X,
82 6HVALC UNIT MEAN TIME(MNS) REMOVAL PER,3X,
83 6H0HRTS COND REPAIR TIME PROCUR LEAD QPA,/
84 6 5X,11HUNIT NSN,12X,
85 6 14HCODE COST,2X,12H1678H FLY NBS,3X,
86 6 40HDATE RATE BASE DEPOT TIME(MONTHS) NBS )
87 WRITE(NOUT,320)IDLRU,ELC,CLRU,INTSD,TRATE,THRS,
88 6 CONB,DTLRU,DTLBU,PLT,APP
89 320 FORMAT(5H0LRU ,124H1679,0,142,1,9X,16,5,9X,295,2,
90 6 16,06277,0,8X,13,0/NN SRU)
91 IF(YNMTS;LE,0)GO TO 322
92 CONL=CONL+THRS
93 PRT=CONL*PRTF30,*CYAC/THRS
94 DTLRU = DTLRU + APPRT
95 DTLRU=PLT*(THRS-CONL)+DTLRU/YNMTS
96 GO TO 823
97 322 YNMTS=CONL
98 PLT=60,
99 DTLRU=60,
100 323 YMTSD=INTSD/APP
101 GO TO 1
102 GO TO 1
103 GO TO 1
104 GO TO 1
105
106 6
107 6
108 25 DECODE(CARD,125)OSL,APP
109 125 FORMAT(25X)4I2,4X,4E364X,8Y11
110 IF ( DSTOCK ) DECODE(CARD,124) LNWOK

```

25-4 LRU PHASE PROVISIONING AND RELEVANT SCHEDULE

1 01 10-26-79 15.250

```

11      124 FORMAT(5HX,I3)
12      GO TO 1
13
14      C      26-7 LRU FEDERAL STOCK # & UNIQUE IDENT
15      C
16      26 DECODE(CARD,126)IDFEN,NSR,NSRNU
17      124 FORMAT(A24,I2,A16,2X,F8.4)
18      IF(IDLRU.NE.IDFEN)GO TO 1
19      IF(NSRNEW.LT.0)NSRNEW=1
20      WRITE(ROUT,226)FSN
21      226 FORMAT(13X,A16/)
22      GO TO 1
23      27 GO TO 1
24      28 GO TO 1
25      29 GO TO 1
26
27      C      30 GO TO(31,32,33,34,35,36,37,38,39)XC2
28      C
29      C      31-4 SRU BASIC DATA
30      C
31      31 IF(1SWRU.LT.1)GO TO 1050
32      IF ( 127.LT.0 ) GO TO 1040
33      NSRU=NSRU+1
34      IF(NSRU.GT.MAXSRU)GO TO 1050
35      J=J+1
36      1SWRU=1
37      DECODE(CARD,131)IDSRU(NSRU),BUC(NSRU),SPSD(NSRU),SSD(NSRU),
38      & CNU(NSRU),XNTD(NSRU),XNRU(NSRU),CON(NSRU),NAPP;
39      & DTSRU(NSRU),DTSNU(NSRU),SPLE(NSRU),SPLC(NSRU),SPO(NSRU)
40      131 FORMAT(A24,3A1,F7.0,F6.0,F8.4,F9.2,F12.3,F10,A1,A4)
41      *W 1282 INCOMPATIBLE W.D FIELD IN 1P1 SPECIFICATION
42      *W 1222 THE CHARACTERS PRECEDING ABOVE FORMAT ERROR ARE .F8.2.
43      C      ****
44      C      ****
45      SAPP=MAXO(NAPP,1)
46      XRATE=ENOUS*SAPP*400/XNTD(NSRU)
47      IF ( 104.GT.4 ) XRATE = 0.0
48      FGLI(NSRU) = XRATE / XRATE*(1-XRTE)
49      FSUM=FSUM+XRATE
50      WRITE(ROUT,231)J,IDSRU(NSRU),SPLC(NSRU),CON(NSRU),
51      & XNTD(NSRU),XRATE,XRTE(NSRU),CON(NSRU),DTSRU(NSRU),
52      & DTSNU(NSRU),SPLT(NSRU),S6K9
53      231 FORMAT(1X,I2,2X,A24,A1,F9.0,F12.4,F9X,F8.5,F4X,2X,I2,
54      & F6.0,2F7.0,8X,F3.0)
55      XNTD(NSRU)=XNTD(NSRU)/(XNRU+APP)
56      IF(XNRU(NSRU).LT.0.1)GO TO 232
57      CON(NSRU)=CON(NSRU)*XRTE(NSRU)
58      SPLT(NSRU)=CON(NSRU)*SPLT(NSRU)*30.*CFAC/XNRU(NSRU)
59      DTSRU(NSRU) = DTSRU(NSRU) + APPRT
60      DTSRU(NSRU)=SPLT(NSRU)+(XRTE(NSRU)*CON(NSRU))
61      & DTSRU(NSRU)/XRTE(NSRU)
62      GO TO 1

```

INDEX THE DATA

```

CSET = CERV(1)
IF ( NOUNOD ) CSET = CERV
CALL INITAL
RETURN
END

```

7 MEMORY EXPANDED, USE SLIMITS OR CORES OPTION FOR NEXT RUN

```

61      232 XNRTS(NSRUS)*CON(NSRUS)
62      SBLT(NSRUS)=60.
63      DNTSRU(NSRUS)=60.
64      GO TO 1
65      32 GO TO 1
66      33 GO TO 1
67      34 GO TO 1
68      C
69      C      35- SRU PHASE PROVISIONING & DELIVERY SCHEDULE
70      C
71      35 DECODE(CARD,295)ID
72      235 FORMAT(A24)
73      IF(ID,NE,IDSRU(NSRUS))GO TO 1
74      DECODE(CARD,135)(QSS(NSRUS,X))X=1,8,(OFF(NSRUS,X),X=1,8)
75      135 FORMAT(25X,4I2,4X,4E22.12,8I1)
76      IF ( DSTOCK ) DECODE(CARD,136) INVRK(NSRUS),BCTPR(NSRUS)
77      134 FORMAT(5X)I2,FB,01
78      GO TO 1
79      C
80      C      36- SRU FEDERAL STOCK NUMBER
81      C
82      36 DECODE(CARD,136)IDFNS
83      136 FORMAT(A24)
84      IF(IDFNS,NE,IDSRU(NSRUS))GO TO 1
85      DECODE(CARD,236)STFNS(NSRUS)
86      236 FORMAT(25X,A16)
87      IF ( STFNS(NSRUS),NE,BLANK16 ) WRITE(ROUT,839) STFNS(NSRUS)
88      336 FORMAT(13X)A16)
89      GO TO 1
90      37 GO TO 1
91      38 GO TO 1
92      39 GO TO 1
93      C
94      C      40(M1)= MODPSE OF X SRU(GO TO SRU DECODE)
95      C
96      40 NMODS = NMODS + 1
97      GO TO 80
98      50 GO TO 1
99      60 GO TO 1
100     70 GO TO 1
101     80 GO TO 1
102     999 IEQJ = 1
103     1000 IF ( IEQJ.LE.0 ) GO TO 1010
104     IEHNRUM0
105     GO TO 3000
106     1010 WRITE(8,1020)
107     1020 FORMAT(' MISSING 97 CARD - ABORT JOB')
108     STOP
109     1030 WRITE(8,1040)
110     1040 FORMAT('NOFLYING NOUR CARD INCOMPLETE')
111     STOP
112     1050 WRITE(ROUT,1055)CARD

```

12 01 10628-79 12,159

```

1      SUBROUTINE OUTPUT
2      COMMON /IOBT/ NORTZ,ENMOD,INLCOM,XBASEX,ENGINE,DSTOCK
3      LOGICAL ENMOD,INLCOM,XBASEX,ENGINE,DSTOCK
4      COMMON /XYSX/ PLTMIN,EXTMAX,X1OMIN,X1OMAX,XENSLON
5
6      COMMON /CHASZ/ HDS,ISRE,INO,ISR,ISLNU,ISSEN,FSN,SFSN,ALC,SALC
7      & XDCNS,TCT,TMT
8      CHARACTER HDG*69,IMOR9,IDLNU*24,ISRU*24(45),FSN*16,SFSN*16(45),
9      & ALC*2,SALC*1(45)/ZPERD(9),KEY*6,TCT*6(3),TMT*5(3)
10
11     COMMON /BASE/ NBASIS,NSRUS,FM,PLYTOT,OST,BETA,BETA1,PRD,BPBD,
12     & BREF,BETOC,BSTAB,BSTOLD,BSTNEW,BSTOB,BSTOB
13     DIMENSION UN(30),OST(60),IBREF(30)
14
15     COMMON /LRUS/ CLRUS,ENTBD,TNETS,BRZLRU,BTSLRU,BLT,LNORK
16
17     COMMON /SRUB/ CSRU,OMEN,CST,ENMOD,XMTD,NETS,BTSLRU,BTSLRU,BPLT,
18     & LNORK,PCTNET
19     DIMENSION DSRU(45),XMTD(45),TNETS(45),BRZSRU(45),BTSLRU(45),
20     & BPLT(45),LNORK(45),PCTNET(45)
21
22     COMMON /SAVE/ ADE,ZELE,HZ,PETHZ,TZ,XPHON,BZERON
23
24     COMMON /STCK/ DSTCK,STCK,RENOVL,BACK,BRCK
25     INTEGER DSTCK(45,3),STCK(45,30,3)
26     DIMENSION BACK(45,30,3),RENOVL(30),BACK(30)
27
28     DIMENSION ADZ(30),FSLZ(30),HZ(30),YZ(30),BZ(30),TZ(30),
29     & XPHON(30,10)
30
31     COMMON /HUND/ LTOT,SLRU,SLRU,SLRU,NSRU,NSRU,NSRU,NSRU,MAXSTK
32     & NSST,BRUMAX,BSTMAX,MAXLRU,MTAB,SLIN,3FONMABS,LRUS,LRPT
33     INTEGER SLRP(30),NSRU(45,30),NSRU(45),NSRU(45),LSRUS(2),
34     & LRPT(2)
35
36     COMMON /HOUT/ TARGST,HOBT,NETS,XPENT,XPHON,XPLE
37
38     COMMON /BOUT/ NPTST,NSST,NUDS,TCOST,BAKO,BATE,BBOD,BMIN,XBMIN
39     & BZEN,BEOL,BZIN,BACK
40     DIMENSION BOD(30),TCOST(3),BAKO(30),BATE(30),BACK(45)
41
42     COMMON /XPHON/ DELTA,MEAN,HLON,ADBD,BMEAN,FDB,BDB,BMEANM
43     DIMENSION DELTA(1000),BDB(45),BDB(45),BMEAN(45)
44
45     COMMON /LCOM/ PSGLZ,BGLT
46
47     COMMON /PEP/ PSGLZ,BGLT
48     COMMON /PEP/ PSGLZ,BGLT
49     COMMON /PEP/ PSGLZ,BGLT
50     COMMON /PEP/ PSGLZ,BGLT
51     COMMON /PEP/ PSGLZ,BGLT
52     COMMON /PEP/ PSGLZ,BGLT
53     COMMON /PEP/ PSGLZ,BGLT
54     COMMON /PEP/ PSGLZ,BGLT
55     COMMON /PEP/ PSGLZ,BGLT
56     COMMON /PEP/ PSGLZ,BGLT
57     COMMON /PEP/ PSGLZ,BGLT
58     COMMON /PEP/ PSGLZ,BGLT
59     COMMON /PEP/ PSGLZ,BGLT
60     COMMON /PEP/ PSGLZ,BGLT
61     COMMON /PEP/ PSGLZ,BGLT
62     COMMON /PEP/ PSGLZ,BGLT
63     COMMON /PEP/ PSGLZ,BGLT
64     COMMON /PEP/ PSGLZ,BGLT
65     COMMON /PEP/ PSGLZ,BGLT
66     COMMON /PEP/ PSGLZ,BGLT
67     COMMON /PEP/ PSGLZ,BGLT
68     COMMON /PEP/ PSGLZ,BGLT
69     COMMON /PEP/ PSGLZ,BGLT
70     COMMON /PEP/ PSGLZ,BGLT
71     COMMON /PEP/ PSGLZ,BGLT
72     COMMON /PEP/ PSGLZ,BGLT
73     COMMON /PEP/ PSGLZ,BGLT
74     COMMON /PEP/ PSGLZ,BGLT
75     COMMON /PEP/ PSGLZ,BGLT
76     COMMON /PEP/ PSGLZ,BGLT
77     COMMON /PEP/ PSGLZ,BGLT
78     COMMON /PEP/ PSGLZ,BGLT
79     COMMON /PEP/ PSGLZ,BGLT
80     COMMON /PEP/ PSGLZ,BGLT
81     COMMON /PEP/ PSGLZ,BGLT
82     COMMON /PEP/ PSGLZ,BGLT
83     COMMON /PEP/ PSGLZ,BGLT
84     COMMON /PEP/ PSGLZ,BGLT
85     COMMON /PEP/ PSGLZ,BGLT
86     COMMON /PEP/ PSGLZ,BGLT
87     COMMON /PEP/ PSGLZ,BGLT
88     COMMON /PEP/ PSGLZ,BGLT
89     COMMON /PEP/ PSGLZ,BGLT
90     COMMON /PEP/ PSGLZ,BGLT
91     COMMON /PEP/ PSGLZ,BGLT
92     COMMON /PEP/ PSGLZ,BGLT
93     COMMON /PEP/ PSGLZ,BGLT
94     COMMON /PEP/ PSGLZ,BGLT
95     COMMON /PEP/ PSGLZ,BGLT
96     COMMON /PEP/ PSGLZ,BGLT
97     COMMON /PEP/ PSGLZ,BGLT
98     COMMON /PEP/ PSGLZ,BGLT
99     COMMON /PEP/ PSGLZ,BGLT
100    COMMON /PEP/ PSGLZ,BGLT

```

TWOIND/OUTPUT

[illegible]

01 01 10628-79 18,259

```

105 IF (NLNUB,NE,EDPIFE) 60 TO 30
106 NLNUB=EDPIFE
107 KC=(NRBOT(NLNUB)+.99)*ELT/DRISNU
108 KD=NRNUB-KC
109 IF (LD,LT,0) LD=0
110 30 CONTINUE
111 DO 40 IBS=1,NBASES
112 EDPIFE=EDPIFE(IBS) + 0.5
113 IF (NLNUB(IBS).LT.EDPIFE) NLNUB(IBS)=EDPIFE
114 40 CONTINUE
115 CEN---*****CHECK 3BU BOUNDS---
116 DO 50 IBS=1,NBSUBS
117 EDPIFE=EDPIFE(IBS) + 0.5
118 IF (NSRUB(IBS).LT.EDPIFE) NSRUB(IBS)=EDPIFE
119 DO 70 IBS=1,NBASES
120 EDPIFE=EDPIFE(IBS) + 0.5
121 IF (NSRUB(IBS).LT.EDPIFE) NSRUB(IBS)=EDPIFE
122 70 CONTINUE
123 50 CONTINUE
124 90 CONTINUE
125 CEN---*****END OF LEVELS CHECK---
126 C
127 WRITE(ROUT,5040) IDBS,INO,IVR
128 5040 FORMAT(1X,17,12(7H---),1/10X,
129 & 8HSUMMARY OF REQUIREMENTS, DISTRIBUTIONS AND COSTS //
130 & 8X,12HWORK PART:14X,3HALL,13X,32HREQUIREMENTS
131 & ,8X,5HDATE,12,45,22,22/8X,16HUNIT NUMBER,9X,
132 & 40HCOSE COST,6X,20HTOTL COST COND BASES )
133 C
134 WRITE(ROUT,5050) IDLNU,ELC,CTL,LTOR,LD,LCF
135 & (NRNUB(IBS) IBS=2,NBASES)
136 5050 FORMAT(5HOLRU ,A24,A10,17,17,216,2X,2313/62X,2313)
137 C
138 IF (IEXLE,0) WRITE(IFILE,5060) IDLNU,FSN,ELC,CTL,LTOT,
139 & LD,LC,IDLNU(IBS) IBS=1,NBASES)
140 5060 FORMAT(1X,2H21 ,A24,A10,17,17,216,2X,2313)
141 WRITE(ROUT,5070)
142 5070 FORMAT(4H 3BU)
143 DO 5100 IBS=1,NBSUBS
144 KC=(ELBOT(NSRUB(IBS))+.99)*ELT/DRISNU(IBS)
145 KD=NSRUB(IBS)-KC
146 C
147 IF (IEXLE,0)
148 & WRITE(IFILE,5080) IDLNU(IBS),FSN(IBS),ELC(IBS),CTL(IBS),
149 & NSRUB(IBS),KC,KC,(NSRUB(IBS) IBS=1,NBASES)
150 5080 FORMAT(1X,2H21 ,A24,A10,17,17,216,2X,2313)
151 C
152 IF (IEXLE,0) WRITE(ROUT,5105) XBACK(3BU)
153 5105 FORMAT(1X,17,12(7H---),1/10X,
154 5100 WRITE(ROUT,5110) IBS,IDLNU(IBS),ELC(IBS),CTL(IBS),NSRUB(IBS),
155 & KD,KC,(NSRUB(IBS) IBS=1,NBASES)
156 5110 FORMAT(1X,2H21 ,A24,A10,17,17,216,2X,2313/62X,2313)

```


: 01 10-26-79 10:159

```

57      C
58      JCT = CT
59      DO 5120 I = 1,4
60      IF ( JCT ) 5120,5120,5117
61      5117 ICT = SCT * ( JCT/1000 ) * 1000
62      DO 5118 J = 1,3
63      KCT(4-J) = ICT * (JCT/10) * 10
64      5118 ICT = ICT / 10
65      JCT = SCT / 1000
66      IF ( I=4 ) 5120,5120,5119
67      5119 FMT(2) = TCT(I=4)
68      GO TO 5122
69      5120 WRITE(ROUT,5121) KCT
70      5121 FORMAT(//40X,10H*** TOTAL COST $,11X,2X1)
71      GO TO 5128
72      5122 WRITE(ROUT,FMT) KCT
73      5128 CONTINUE
74      5129 WRITE(ROUT,5130) XNIN
75      5130 FORMAT(40X,10H*** RECORDERS #,214.6)
76      WZEROS = 0.200
77      FFLT=0.0
78      DO 5132 IBS = 1,NBSSES
79      FELT=FELT+FILE(IBS)*RHS(2BS)
80      WZEROS = WZEROS + WZEROS(IBS,1)*RH(IBS)
81      5132 CONTINUE
82      WZEROS = 100.0 * WZEROS / FFLTOT
83      FELT=FELT/FLYTOT
84      WRITE(ROUT,5133) 100.0*FELT
85      5133 FORMAT(40X,10H*** BILL RATE X ,214.2)
86      WRITE(ROUT,5134) WZEROS
87      5134 FORMAT(40X,10H*** LEADY RATE X ,214.2)
88      WRITE(ROUT,5140) BBSZ
89      5140 FORMAT(40X,10H*** BUT SUPP ORG ,214.5)
90      IF (IPRNT,LE,0) GO TO 5205
91      C
92      C      GENERATE CONDITIONAL SRU WAIT TIMES FOR BCSH INPUT
93      C
94      IF ( .NOT.INLCOM ) GO TO 5169
95      C
96      DO 5142 SRU = 1,NBSRUS
97      CALL ABSENSE( PDD(SRU),SRUB(SRU),DNBSRUS(SRU) )
98      5141 DELAY(SRU) = DELTA(SRUB(SRU)) * DRTSRU(SRU)
99      C
100     DO 5148 IBS = 1,NBSSES
101     CADZ(IBS) = 0.0
102     DO 5148 SRU = 1,NBSRUS
103     THJ = (17-INPTS(SRU)) * DRTSRU(SRU)
104     + XNIN(SRU) * (DST(IBS)*DELAY(SRU))
105     PX = WBD(SRU) * PDD(SRU)
106     P4 = 1. - PDD(SRU)
107     R4 = RBD(SRU) - 1.
108     PCUM = 0.0

```

10 02 40025-79 1A1359

```

109      STK = 0.0
110
111      C
112      5142 RCUN = RCUN + PX
113      IF ( SUBTOT, NSRUB(ZRU2TSS) ) GO TO 5143
114      STK = STK + 1.0
115      PX = PX + 1.0 * (STK-1) / STK
116      GO TO 5142
117
118      C
119      5143 CONTINUE
120      5143 CADZ(ZSS) = CADZ(ZSS) + RSOLZ(ZSU) * (1.0 / RCUN) * STK
121
122      C
123      CADZ(ZSS) = CADZ(ZSS) / RSOLZ
124      5144 CONTINUE
125
126      C
127      5145 CONTINUE
128
129      DO 5150 NKE = 2, 10
130      DO 5150 ZSS = 1, NBASES
131      K = NLRUB(ZSS) + 1
132      NZ(ZSS) = (X * RE(ZSS) - 1.0) * (1.0 - PE(ZSS)) * NZ(ZSS) / X
133      KPROB(ZSS, NKE) = NZ(ZSS)
134      NSRUB(ZSS) = NLRUB(ZSS) + 1
135
136      5150 CONTINUE
137
138      5160 CONTINUE
139      WRITE(ROUT, 5170)
140
141      5170 FORMAT('X', 18(6H+*****), 26H ***** PROBABILITY OF X,
142      & 26H BACKORDERS AT THE POINT IN TIME ***** FALL,
143      & 26H LRU WAIT TIME EXPECTED TOTAL )
144      WRITE(ROUT, 5180) (ZSS=1, 9)
145
146      5180 FORMAT('LNU BASE 100, 9450 X0, I1) X0X0
147      & 44H RATE FOR REEDED SRU LRU REUPPLY TIME,
148      & / 9H ***** 10 (6H *****), 24H ***** (LNU+*****))
149      IF ( INLCOM ) WRITE(ROUT, 5195)
150
151      5195 FORMAT('99X, 14H (GIVEN WAIT) )
152      DO 5200 ZSS = 1, NBASES
153      WRITE(ROUT, 5190) ZSS, KPROB(ZSS, ZSS), ZSS=1, 10, RELZ(ZSS),
154      & ADZ(ZSS) / PE(ZSS)
155      IF ( INLCOM ) WRITE(ROUT, 5195) CADZ(ZSS)
156
157      5195 FORMAT('1H+19QX, 1H(20H(1.04R) )
158      5190 FORMAT('1X, 25.3X, 10F5.2, 27F5.2, 13(2.0X, 19(2)
159
160      5200 CONTINUE
161      CALL PTIME(ETIME)
162      WRITE(ROUT, 5202) ETIME
163
164      5202 FORMAT('0ELARSED CONNTE TIME =', 29, ' N16ISEC0, ')
165
166      5205 NSTS=NSTS+1
167      BAKO(NSTS)=KPNIN
168      BUDG(NSTS)=CF
169      BATE(NSTS)=FILT
170
171      C-----RESET LEVELS TO THE INPUT VALUES
172      5200 CONTINUE
173      NSRUB=NLRUB
174      DO 5220 ZSU=1, NSRUB

```

: 01 10626-79 1R1259

```
51      MSRUD(IRU)=MSRUD(IRU)
52      DO 5025 IBS=1,MSRUDS
53          MSRUD(IRU,IBS)=MSRUD(IRU,IBS)
54          MLRUD(IBS)=MLRUD(IBS)
55      5015 CONTINUE
56      5020 CONTINUE
57      C-----
58      RETURN
59      END
```

01 10-20-79 10.803 13

*BUDGET=LINE/ORD/COMBIN.O(BCD,NOBGT
 *COMBIN.S

C COMBINS
 C COMPILED FILE IS MODNETRIC/ORDORD

C
 DIMENSION BUD(200,20),BACK(200,20),NSV(400,43),NRPL(200)X
 4 NL(200),ID(200,6),SL(200)
 CHARACTER STAR*3/3H**/,BLANK(3/3H /,LEU*6,LINE*3(200)
 CHARACTER JBLANK*1/1H /,JVLNGT1
 CHARACTER NAME*6,JLEU*6
 DIMENSION IDORD(9)

C
 CALL ISNT(XNUMB,IDORD)
 CALL DATE(1DATE)
 10 READ(4:20,END=9,END=9)NAME
 IF(NAME(1).EQ."LAST ")GO TO 16
 BACKSPACE 4
 WRITE(6,80)1DATE,ISNUMB,ISORD
 20 FORMAT(12,A6)
 30 FORMAT(1" COMBINS",1X,16,1X,15F1X,9A6)
 40 FORMAT(1X,"LAST ")
 WRITE(6,80)
 9 CALL DETECH(6,1STAT, 7

C
 MAXINT=200
 NOUT=6
 NSW=1

C-----LINE CHANGES

BMIN=0.
 BMAX=1.810

C-----END CHANGES

50 FORMAT(2F10,0)
 IF(BMAX.EQ.0.)BMAX=1.810
 WRITE(NOUT,60)BMIN,BMAX
 60 FORMAT(1"MIN BUDGET="F11,0,F50,"MAX BUDGET="F11,0)
 WRITE(NOUT,70)
 70 FORMAT(1NO,24X,4NUNIT,4X,6NBUDGET,6X,10NBACKORDERS/
 8 22NODATA SCINT SET NO. 4)

ISNT=1
 IPT=0
 DO 80 I=1,80
 80 LINE(I)=BLANK
 90 READ(5:100,END=200)JFLAG,JLEUBUDGET,BACKOR
 100 FORMAT(A1,A6,F11,0,F14,4)
 IF(JFLAG.EQ.1)GO TO 160
 LEU=0LEU
 IF(IPT(10,0)GO TO 110
 IF(BUDGET.LE.BUD(ISNT,IPT).OR.BACKOR.GE.BACK(ISNT,IPT))GO TO 140
 108 IF(IPT(1,2)GO TO 110
 S1=(BACK(ISNT,IPT=1)-BACK(IAND,IPT)) /
 (BUD(ISNT,IPT)-BUD(ISNT,IPT=1))
 8 S2=(BACK(ISNT,IPT)-BACKOR)/(BUDGET-BUD(ISNT,IPT))

COMBIN

01 10-20-79 11.505 28

```

      IF(S2.LT.S1)GOTO 110
      WRITE(NOUT,155)LRU,BUD(ISET,IPT),BACK(ISET,IPT)
155  FORMAT(60X,"NON-CONVEX PT SKIPPED ",A6,F10.0,F14.4)
      IPT=IPT+1
      GOTO 108
110  IPT=IPT+1
      BUD(ISET,IPT)=NBUDGET
      BACK(ISET,IPT)=NBCKOR
      DECODE(LRU,180)(ID(ISET,J),J=1,6)
130  FORMAT(60X)
      GOTO 90
140  WRITE(NOUT,180)LRU,BUDGET,BCKOR
150  FORMAT(60X,"ERROR ",A6,F10.0,F14.4)
      GOTO 90
160  NPPLRU(ISET)*IPT
      WRITE(NOUT,180)(I,LRU,BUD(ISET,I),BACK(ISET,I),I=1,IPT)
120  FORMAT(24X,IS,1XIA6,F10.0,F14.4)
      IF(IPT.GT.1)GO TO 180
      KSW=0
      WRITE(NOUT,170)
170  FORMAT(60X," ERROR ,MORE THAN 1 POINT REQUIRED ")
180  ISET=ISET+1
      IF(ISET.GT.MAXSET)GOTO 370
      WRITE(NOUT,180)ISET
190  FORMAT(19X)DATA POINT SET NO.,IS)
      IPT=0
      GOTO 90
200  WRITE(NOUT,210)
210  FORMAT(12X)NO OF DATA)
      IF(KSW.EQ.1)GO TO 230
      WRITE(NOUT,220)
220  FORMAT(24X)NO ERRORS IN INPUT PINE)
      WRITE(NOUT,220)
      STOP
230  NLRU=ISET-1
      NA=1
      NST=32
240  NST=MIN0(NST,NLRU)
      TCOST=0.
      TBCK=0.
      WRITE(NOUT,250)
250  FORMAT(18X,15X)WORK UNIT CODES)
      DO 26 J=1,6
260  WRITE(NOUT,270)(ID(I,J),I=NX,NST)
270  FORMAT(38X,38(2X,A1))
      DO 28 I=1,NLRU
      NL(I)=1
      SL(I)=(BACK(I,1)-BACK(I,2)) / (BUD(I,2)-BUD(I,1))
      TBCK=TBCK+BACK(I,1)
280  TCOST=TCOST+BUD(I,1)
      WRITE(NOUT,290)TCOST,TBCK
290  FORMAT(28X)BUDGET      BACKORNB      B80/

```

01 10-20-79 10.509 15

```

      1470 IF(0.0.F10.4.F20X,1470)GOTO 300
300 ISW=0
      IF(TCONT.GT.BMAX)GOTO 360
      BSO=-1.
      DO 310 I=1,NLNU
      IF(NL(I).GE.NPPLBU(I))GOTO 310
      IF(SL(I).LE.BSO)GOTO 310
      ISW=1
      BSO=SL(I)
      N=I
310 CONTINUE
      IF(ISW.EQ.0)GOTO 360
      KOLD=NL(N)
      NL(N)=SL(N)+1
      KNEW=NL(N)
      IF(KNEW.NT.NPPLBU(N))GOTO 320
      NL(N)=99
      GOTO 380
320 SL(N)=(BACK(N,KNEW)-BACK(N,KNEW+1))/(BUD(N,KNEW+1)-BUD(N,KNEW))
330 TCOST=TCOST-BUD(N,KOLD)+BUD(N,KNEW)
      TBACK=TBACK-BACK(N,KOLD)+BACK(N,KNEW)
      IF(TCONT.LT.BMIN)GOTO 300
      LINE(N)=NSTAR
      WRITE(ROUT,340)(LINE(I),I=NX,NST)
340 FORMAT(39X,3E13)
      LINE(N)=NLANN
      WRITE(ROUT,350) TCOST,TBACK,BSO,(NL(I),I=NA,NST)
350 FORMAT(1E,10.0,10.4,E12.5,3E13)
      GOTO 360
360 IF(NST.GE.NLNU)STOP
      NA=NST+1
      NST=NA+31
      GOTO 260
370 WRITE(ROUT,380)
380 FORMAT(20HODIMENSION TOO SMALL )
      STOP
      END

```

1470 EQUALITY OR NON-EQUALITY COMPARISON MAY NOT BE MEANINGFUL IN LOGICAL IF EX

REFERENCES

1. Demmy, W. Steven, RIME: The Recoverable Item Management Evaluator, Volume I: Model Description, TR-80-01, Decision Systems, 3575 Charlene Drive, Dayton, Ohio 45432, May 1980, 153 pp.
2. Demmy, W. Steven, RIME: The Recoverable Item Management Evaluation Volume II, Program Listings and Narratives, TR-80-02, Decision Systems, 3575 Charlene Drive, Dayton, Ohio 45432, May 1980, 298 pp.
3. Demmy, W. Steven, An Empirical Evaluation of Proposed Stockage Policies for Recoverable Item Management, TR-80-03, Decision Systems, 3575 Charlene Drive, Dayton, Ohio 45432, May 1980, 173 pp.
4. Demmy, W. Steven and Victor J. Presutti, Jr., Multi-Echelon Inventory Theory in the Air Force Logistics Command, Working Paper 76-3011-27, Department of Management, Wright State University, Dayton, Ohio 45435.
5. Muckstadt, John A., "A Model for a Multi-Item, Multi-Echelon, Multi-Indenture Inventory System", Management Science, v20, n4, December, 1973, pp. 472-481.